NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DETECTION OF ACTIVE TOPOLOGY PROBING DECEPTION

by

Weiyou Nicholas Phua

September 2015

Thesis Advisor:	Robert Beverly
Second Reader:	Justin P. Rohrer

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE 09-25-2015	3. REPORT TYPE AND DATES COVERED Master's Thesis 09-29-2014 to 09-25-2015

4. TITLE AND SUBTITLE DETECTION OF ACTIVE TOPOLOGY PROBING DECEPTION	5. FUNDING NUMBERS H98230221650
6. AUTHOR(S) Weiyou Nicholas Phua	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

For all purposes and intents, being able to infer the topology of a network is crucial to both operators and adversaries alike. Traceroute is a common active probing technique but it may be subverted by deceptive responses. We identify possible inconsistencies in traceroute deception systems, and endeavor to find potential deception in the historic IPv4 Routed /24 Topology Dataset from the Center for Applied Internet Data Analysis (CAIDA). Our results show three major patterns in 2013 and 2014 that exhibited instances of inconsistencies matching the techniques in our methodology. In addition to analyzing the historic dataset, we evaluate three cases of traceroute manipulation in the wild. These case studies include The Pirate Bay (TPB) server supposedly residing in North Korea, the Star Wars- and Christmas Carol-themed gags involving customized Domain Name System (DNS) names, and the experimental DeTracer at the Naval Postgraduate School (NPS). In the TPB case, we discovered extensive and long-running deception in the /24 subnet. We find intriguing patterns in the gag traceroutes and fake topologies from the DeTracer for which we may use to improve our filtering process. In all, the findings will aid future operations in verifying inferred network topologies from traceroutes.

14. SUBJECT TERMS network topology, network deception, traceroute	15. NUMBER OF PAGES 95
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

THIS PAGE INTENTIONALLY LEFT BLANK

DETECTION OF ACTIVE TOPOLOGY PROBING DECEPTION

Weiyou Nicholas Phua
Civilian, Singapore
B.Eng. Computer Science, Nanyang Technological University, 2009

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE
from the

NAVAL POSTGRADUATE SCHOOL
September 2015

Author: Weiyou Nicholas Phua

Approved by: Robert Beverly
Thesis Advisor

Justin P. Rohrer
Second Reader

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

For all purposes and intents, being able to infer the topology of a network is crucial to both operators and adversaries alike. Traceroute is a common active probing technique but it may be subverted by deceptive responses. We identify possible inconsistencies in traceroute deception systems, and endeavor to find potential deception in the historic IPv4 Routed /24 Topology Dataset from the Center for Applied Internet Data Analysis (CAIDA). Our results show three major patterns in 2013 and 2014 that exhibited instances of inconsistencies matching the techniques in our methodology. In addition to analyzing the historic dataset, we evaluate three cases of traceroute manipulation in the wild. These case studies include The Pirate Bay (TPB) server supposedly residing in North Korea, the Star Wars- and Christmas Carol-themed gags involving customized Domain Name System (DNS) names, and the experimental DeTracer at the Naval Postgraduate School (NPS). In the TPB case, we discovered extensive and long-running deception in the /24 subnet. We find intriguing patterns in the gag traceroutes and fake topologies from the DeTracer for which we may use to improve our filtering process. In all, the findings will aid future operations in verifying inferred network topologies from traceroutes.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

Ark	Archipelago
AS	Autonomous System
ASN	Autonomous System Number
BGP	Border Gateway Protocol
CAIDA	Center for Applied Internet Data Analysis
CDN	Content Delivery Network
CMAND	Center for Measurement and Analysis of Network Data
CND	Computer Network Defense
DNS	Domain Name System
DOD	Department of Defense
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPID	IPv4 Identification
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IRC	Internet Relay Chat
ISP	Internet Service Provider
IPS	Intrusion Prevention System
LSE	Label Stack Entry

MPLS	Multiprotocol Label Switching
NPS	Naval Postgraduate School
NIST	National Institute of Standards and Technology
RFC	Request for Comments
RIB	Routing Information Base
RIR	Regional Internet Registries
RTT	Round Trip Times
PTR	Pointer Record
TCP	Transmission Control Protocol
TTL	Time-to-Live
UDP	User Datagram Protocol
US	United States
USG	United States Government
VRF	Virtual Routing and Forwarding

Acknowledgments

I would like to thank Dr. Robert Beverly for his patience during the research and thesis writing process. His deep knowledge on networking topics has been most useful in guiding me along the right courses of action in my research. I am grateful for his emphasis on precision and clarity to help me focus on the important aspects of this thesis. I would also like to thank Dr. Justin Rohrer for his commitment and support in enabling me to work on my research without interruptions.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction

Active network probing is a means of discovering a target network's configuration and topology. Many networks and organizations block outside probing for privacy or competitive reasons. This thesis considers an alternate defensive strategy: topology deception. Specifically, we seek to discover properties of deceptive topology countermeasures that hint at their existence, and exploit those properties to discover the prevalence of such systems across the Internet.

According to the definition used in the ATIS Telecom Glossary [1], the network topology is the "specific physical, i.e., real, or logical, i.e., virtual, arrangement of the elements of a network." We consider the elements in the target network like computers, routers, servers, and switches. For instance, a typical active network topology probing operation will enable the creation of a human-readable visual network map.

1.1 Network Topology Probing

Reconnaissance of an adversary's operating areas is a key initial element when defining a concept of operations for future cyber missions. Likewise, in a world interconnected by high-speed networks and shared transmission hubs, intelligence gained from an organization's network topology could allow malicious actors to prioritize their mission objectives and identify especially weak or critical points in the network.

Knowledge of the network topology has benefits to internal and external users of the network. The network administrators might be interested in identifying the traffic bottlenecks and perform rerouting to ease congestion. In like manner, content providers desire optimal delivery of paid content and will seek to find alternative paths to their customers. Adversaries could discover network elements of high utilization to be possible single points of failure and carry out denial-of-service attacks. They may consider injecting a network tap into those elements to become potential traffic eavesdropping points. They may also compromise a border machine and use it as a pivot point to attack other machines on the same network, in an effort to bypass firewall restrictions on external connections.

A common active probing technique is to perform a traceroute to a target network destination in order to determine the forward path (sequence of router interfaces from the source to the destination).

1.2 Probing Countermeasures

A defender can elect to either deny active probing with network filters such as firewalls, or deceive incoming probes with falsified data so as to present a fictitious network. With firewalls, the defender is able to filter incoming probes but some diagnostics capabilities will be lost in the event of network outages. A balance between isolation and troubleshooting capability has to be found for such configurations. On the other hand, data representing a false network topology could be presented to the probes by simulation or using actual decoy hardware.

An alternative decoy technology available is the honeypot. Honeypots and honeynets are canonical examples of decoys that are frequently used for deception and intelligence gathering. Spitzner [2] states that a honeypot is a "security resource whose value lies in being probed, attacked or compromised." A honeynet consists of two or more honeypots in the same network. A honeypot appears to be a legitimate part of the network and lures would-be attackers to compromise it. It is a continuously monitored, isolated system made up of physical or virtual components. In our case, honeypots may be used for deceiving probes and alerting the defender if necessary.

A documented case of employing decoy hardware was the "Star Wars Traceroute" advertised on the Internet by Werber in 2013 [3]. He deployed two Cisco 1841 Integrated Services Routers with Virtual Routing and Forwarding (VRF) [4] and allowed traceroute traffic to be redirected into the two-router chain, generating a custom themed message when the IP addresses were resolved.

In the case of traceroute probing, the defender may manipulate the return traffic with deception outcomes such as hiding legitimate nodes, seeding virtual nodes, and masquerading as other nodes. Critical servers and routers are common examples of legitimate nodes that benefit from obscurity. Virtual nodes could be injected to portray the existence of a smaller internal network and bait possible attackers into investing time and effort to probe them. This enables redirection of an adversary's focus as well as enabling monitoring on those nodes. As the virtual nodes are dedicated decoys, any detected activity is less likely to be considered benign and may be investigated promptly. Nodes masquerading as legitimate nodes may cause confusion if they appear to be connected when no such direct link exists in the real world. An example of such a pretense could be showing adjacent routers being located in countries that do not have direct land or sea communication links to each other.

1.3 Imperfection in Deception

As with all implementations of deception systems, there is a possibility of misconfiguration or flaws, either accidental or deliberate. The implementing party may not have considered all cases and has omitted certain modifications, which leave clues due to an incomplete fabrication. In another case, the deception system may only be designed to counter probes using a specific protocol. An adversary opting to perform active probing using several protocols may observe inconsistencies and become suspicious. In addition to inconsistencies arising from incomplete implementations, sections of the presented deception may not be feasible. Some of the reasons include latency mismatches in deception-supplied and direct packets, impossible physical connections according to the return probes, as well as prior known ground truth.

Aside from imperfections in network topology deception, there may be discrepancies in other deception systems that adversaries may take advantage of to establish authenticity of the presented output. An example is the capitalization of the fact that tarpits such as LaBrea [5] do not return Transmission Control Protocol (TCP) options in its responses, and this revelation can then be used to suggest the existence of a tarpit [6].

Not all network topology deception is designed with the goal to improve network security. There are elaborate gags that control a series of specially crafted, themed Domain Name System (DNS) names such that the eventual output of regular traceroutes tells a story. An example of a such a gag was a reddit post that encouraged users to traceroute to a specific website [7]. The traceroute responses returned Internet Protocol (IP) addresses that, when resolved, were in the form of a Christmas carol.

1.4 Countermeasure Detection

It has been reported that The Pirate Bay, a popular torrent website, performed topology deception by replying to traceroutes with extraneous and false data in 2013 [8]. A blogger by the name of Will [9], wrote that by performing traceroutes of the IP addresses present in his original traceroute response of the website, he discovered the path taken was significantly different. Specifically, he mentioned the Autonomous System (AS) numbers resolved from the IP addresses did not actually link up with each other and this implied that a portion of the reported route was fake. The inconsistency was discovered as a result of deeper probing and elimination of false leads. While this is a known isolated example, this thesis seeks to cast a wide net and generally discover the extent of topology deception present on the wider Internet.

This thesis relies on data that is publicly available to researchers from the Center for Applied Internet Data Analysis (CAIDA). The data are examined for inconsistencies in returning traceroute probes from a target network that may reveal the presence of deception. The scope is limited to past and present traceroute data collected from vantage points across the globe. The prevalence of traceroute deception on the broader Internet is a subsidiary research question.

1.5 Thesis Structure

- Chapter 1 introduces the reader to network topology probing and deceptive countermeasures, and highlights the factors that diminish the quality of deception, leading to the development of specific detection methods.
- Chapter 2 explains concepts related to traceroutes and network topology deception.
- Chapter 3 discusses the methodology for filtering the CAIDA datasets and the techniques used for revealing traceroute deception. It also presents case studies of real-world scenarios of possible deception.
- Chapter 4 reports our findings based on the filtered CAIDA datasets. We also analyze

data on some real-world examples that exhibit interesting network topologies.

- Chapter 5 presents our conclusions as well as future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2: Background and Review of Literature

This chapter introduces the concepts, data sources, and tools used for network topology probing. The deception inconsistencies are explored in greater depth, followed by a review of prior literature on topology deception.

2.1 Traceroute Concepts

The concept of traceroute is to provide insight into the forwarding of IP packets to their destinations. While traceroute [10] is both the name of a utility present in most operating systems as well as a term for the said utility's output, it is now generally considered a technique. Traceroutes are often employed as a diagnostic tool to investigate and troubleshoot network issues. A typical traceroute implementation uses User Datagram Protocol (UDP), Transmission Control Protocol (TCP), or Internet Control Message Protocol (ICMP). Different operating system families tend to use different protocols in their traceroute implementations. For instance, the Microsoft Windows tool "tracert" adopts the use of ICMP [11], while "traceroute" in Linux systems [10] uses UDP as its default protocol. The traceroute technique is heavily reliant on the interpretation of replies from the routers along the path to destination host. As these replies lack any form of authentication and non-repudiation features, they are susceptible to data manipulation for the purposes of masking the network topology, impersonating other hosts, and/or the host operator's preference.

Under IPv4, a traceroute sends a series of probes with incrementing Time-to-Live (TTL) values to a destination. Each router along the forward path decrements the probe packet's TTL value, and if the TTL value is zero, discards the packet and replies to the prober with an ICMP error datagram. As routers provide interfaces for two or more networks to connect to each other, there must be a selection process that decides which interface address to use as the source of the ICMP datagram. Traceroute only reveals the router interfaces receiving and responding to the forwarded probe. Links in between the original prober and its final destination are commonly known as “hops.” The two ICMP message types seen in the responses must be either Time Exceeded or Destination Unreachable. By default, the prober expects a Time Exceeded message for each hop, except the destination for which it expects a Port Unreachable sub-type of the Destination Unreachable type. The Time Exceeded message includes the first 64-bits [12] of the original datagram’s data, allowing the traceroute to compare source addresses and map ICMP responses to the corresponding hops. It has to be noted that this method is causing only the nodes along the path from the original prober to its final destination to report, and does not infer that the return path is always the same. The interpretation of these responses allows the sender to infer the resultant network topology and the per-hop RTT latencies as shown in Figure 2.1 and Figure 2.2.

```
Tracing route from 1.2.3.4 to 5.6.7.8:
Router A [1.2.3.1] 100ms 90ms 100ms
Router B [2.3.4.99] 120ms 120ms 130ms
Router C [3.4.5.99] 150ms 135ms 160ms
Destination [4.5.6.7] 200ms 230ms 220ms
```

Figure 2.1: An example of traceroute output.

A typical traceroute usually sends three probes in succession for each TTL value. In the example result shown in Figure 2.1, the first column reports each of the discovered node with its corresponding interface’s IP address. The round-trip time taken for each of the probes is shown in the last three columns. By default, groups of three probes are sent to compensate for random packet loss and network jitter. The shortest time of the returning probes in the group will be recorded. Figure 2.2 shows an inferred network topology diagram obtained from the results as shown in Figure 2.1. The first group of probes with a TTL of one arriving at Router A will invoke a response through the 1.2.3.1 interface back to the source. The next series of probes of TTL value two will pass Router A and invoke a return response via Router B’s 2.3.4.99 interface as the probe TTLs become zero. Subsequently, the third group of probes will trigger a similar effect at Router C. Finally, the probes arriving at the destination 4.5.6.7 will return an ICMP Destination/Port Unreachable message.

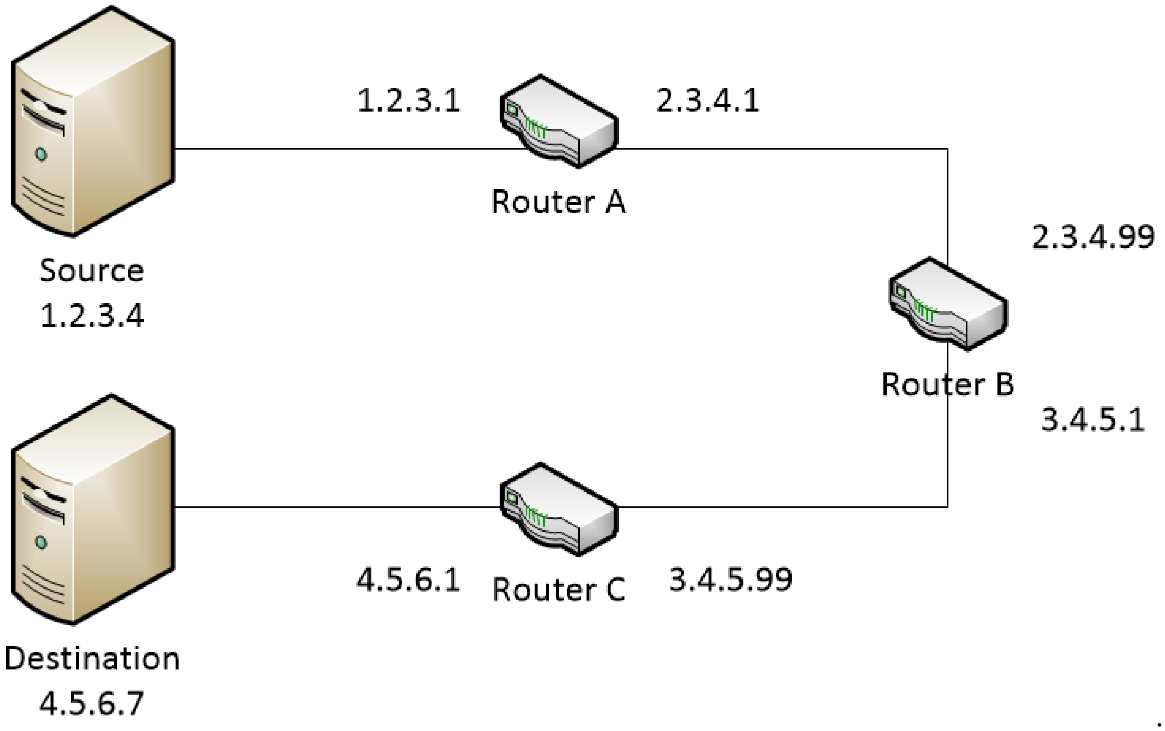


Figure 2.2: Inferred network topology from traceroute.

2.2 Paris Traceroute

Augustin *et al.* [13] has shown that traceroute does not guarantee precise mapping of a network due to the presence of load balancing routers. Routers along the forward path may reside in different networks and have varying data-spreading policies. Their respective network administrators may employ load balancing to redirect data packets to alternative routers based on the current network utilization. The outcome is the existence of alternative routes to the desired destination over the course of the traceroute operation.

There are three main policy categories of load balancing: per-flow, per-packet, and per-destination. The Request for Comments (RFC) 3917 [14] states that “a flow is defined as a set of IP packets passing an observation point in the network during a certain time interval.” Per-flow load balancing looks at the packet header information of each packet, assigns the packet to a flow, and ensures that the router does not disrupt packets belonging to the same flow during the routing process. From the surface, default traceroute probes from the same source are very likely to end up in the same flow for that hop. Unfortunately, the variance in each consecutive distinct probe, for example, varying Destination Port fields, can cause the flow identifier to change. In comparison, per-packet load balancing forwards packets in a round-robin fashion, based on the current network utilization, and increases the likelihood of successive probes being forwarded to different next hop routers. This generates conflicting replies during the affected hops, which leads to multiple branches in the resultant route. Per-destination load balancing is similar to per-flow since it enforces redirection based on the destination IP address. The impact of these load balancing policies is that anomalies such as loops, cycles, diamonds, missing nodes and links, and false links will introduce errors in the topology mapping process.

Augustin *et al.* also introduces Paris Traceroute, which aims to provide a more accurate picture of actual packet flows. It manipulates the probe packet header fields in order for load balancing routers to maintain a constant flow identifier for its probes, and yet be able to match reply packets to their corresponding probe packets. This thesis utilizes results from the Paris Traceroute to determine whether inconsistencies found contribute to actual deception.

2.3 Autonomous Systems

The definition of an AS is "a connected group of one or more IP prefixes run by one or more network operators which has a single and clearly defined routing policy" [15]. These network operators are commonly Internet Service Providers (ISPs) or significantly sized organizations like the Department of Defense (DOD), operating separate connections to various networks. Each AS is to be assigned a globally unique Autonomous System Number (ASN) as an identifier and also used in the exchange of exterior routing information between adjacent ASs [15].

Border Gateway Protocol (BGP) is an inter-AS protocol used to exchange network reachability information between different ASs [16]. A BGP speaking system uses TCP and maintains a table of AS paths for the forward of packets belonging to specific subnets. These AS paths will reveal information about the route in which traceroute responses traverse through. Thus, the physical path may then be analyzed for connections between ASs of non-bordering countries.

2.4 CAIDA Ark and Scamper

CAIDA [17] is “a collaborative undertaking among organizations in the commercial, government, and research sectors aimed at promoting greater cooperation in the engineering and maintenance of a robust, scalable global Internet infrastructure.” CAIDA is also an operator of a globally distributed measurement platform named Archipelago (Ark), with the primary goals of cutting down on effort required for complex large-scale measurements, and promoting community-oriented measurement infrastructure with collaborators being able to run their own vetted measurement tasks [18].

CAIDA hosts measurement datasets collected from 1998 onwards such as the Internet Protocol version 4 (IPv4) Routed /24 Topology Dataset [19]. This dataset consists of scamper warts [20] files categorized according to the probing host and day, which contain traceroute data across all /24 prefixes in the routed IPv4 address space. Luckie [21] designed Scamper to be a scalable and extensible packet-prober for active measurement of the Internet. It includes an implementation of Paris Traceroute for analyzing network topology and performance. The parallel probing nature of Scamper is a good fit for the collection of active measurement data on Ark. Hence, this thesis entails working with the IPv4 Routed /24 Topology Dataset, augmented with IPv4 Routed /24 DNS Names Dataset [22] and the University of Oregon Route Views Archive Project [23] to allow IP address to AS matchups. *Warts* [20] is the native binary output file format of Scamper. It is extensible and able to record considerable detail as well as meta data on each measurement. Scamper also supports output to ASCII text.

The IPv4 dataset was generated in a methodological way as documented [19] in its distribution page. The probing work consisted of continuously sending out scamper probes performing traceroute to all routed /24 networks in the IPv4 address space. The load was divided among three teams of approximately 17—18 Ark monitors in different geographical regions. A destination address would be picked at random from each routed IPv4 /24 prefix on the Internet. Another constraint was the /24 prefix that the address belonged to had to be unique across all monitors within a probing cycle of around 48 hours. Each team’s monitor would then probe the allocated address and store the results in compressed *warts* files.

2.5 IP Geolocation

Successful traceroute responses confer the ability to know more about the physical forward path taken by the probes. Geolocation may be performed on IP addresses through database lookups, measuring delays from multiple vantage points, and obtaining clues from DNS Pointer Records (PTRs) [24]. A *whois* of the IP address queries Regional Internet Registriess (RIRs) will reveal information on the ISP and ASN [25]. As ISPs and ASNs are bound to specific countries, they at least provide a coarse-grained picture of the traversed route. Likewise, the DNS PTR may offer hints on the country origin based on the hostname, if available. The use of delay-based measurements takes advantage of the differences in Round Trip Times (RTT) between geographically distributed vantage points and chosen landmarks. This concept is beneficial to unmasking deception and is explored further in Chapter 3.

The time delay between sending an outgoing probe and receiving its corresponding response on the prober's machine is known as the RTT. It also reveals information on the distance traveled by the probe. The sources of RTT include i) propagation delay, ii) transmission delay, iii) processing and iv) queueing. *Propagation* delay is the time taken for a bit to transit between two routers, while *transmission* delay is the delay resulting from pushing a packet onto the transmission medium. In most cases, the propagation delay is dynamic from varying route and traffic conditions, and dominates the fixed transmission delay, which is dependent on the medium used such as copper and fiber cables, or wireless links. The processing delay comes from time spent by routers performing parsing and processing on the packet headers. Lastly, the queuing delay is dependent on time spent waiting in the buffer of both the sender and receiver devices. It must be noted the delay-based geolocation is relatively inaccurate due to the above sources of delay, as well as routing strategies causing inefficient routes such as circuits and loops. Given that the RTT is calculated on the prober's machine, the country of destination is known, and coupled with signal propagation at the speed of light, the minimum delay possible will be the time taken for light to make a round trip.

2.6 Deception in Computer Security

Computer security encompasses a wide range of security topics centered around data, computers, and computer networks. From the National Institute of Standards and Technology (NIST) definition, computer security comprises of "measures and controls that ensure confidentiality, integrity, and availability of information system assets including hardware, software, firmware, and information being processed, stored, and communicated" [26]. Computer Network Defense (CND) is a term categorizing defensive actions selected against unauthorized activity within computer networks [26]. Yuill states that a deception operation is a planned set of actions taken to mislead hackers and thereby cause them to either embark on or avoid specific actions that aid CND [27]. As an illustration, an offensive mission may apply deception within an adversary's domain or deploy deceptive defensive countermeasures on home ground.

Deception systems could be improved by understanding the nature of deceit and how it works in general. Almeshekah *et al.* elucidated a model for planning and integrating deception in security defenses effectively [28]. They covered a taxonomy of deceitful techniques revolving around the concepts of hiding reality and simulating substitutes. The act of hiding included masking the existence of valued parts, repackaging as a camouflage method, and dazzling as an approach to causing confusion among similar objects. The second concept targeted methods to mimic or imitate, inventing non-existent components, and provide decoys to divert attention. They also outlined the need for understanding and exploiting potential adversaries' biases. The success of a defensive deception system depended on sustainable and plausible alternative perceptions.

Denning described the differences between active and passive cyber defenses using an active and passive air defense analogy [29]. Active defenses include Intrusion Prevention System (IPS), firewalls, honeypots, access controls, among others. The rationale for honeypots and the like is the diversion of attackers' attention to isolated systems where their behavior can be monitored. The result is similar to the act of deflecting approaching missiles and rockets. In contrast, passive cyber defense considers cryptography, monitoring, vulnerability assessment, cybersecurity education, and so forth as examples that emphasizes on making systems tougher to be attacked. In this thesis, we focus on the defensive aspect of deception operations carried out by adversaries in their networks.

Heckman *et al.* summarized some of the possible resources supporting a cyber deception operation and includes honeypots, "fake honeypots," "fake fake honeypots," honeynets, honeyclients, honeytokens, and tarpits [30]. They highlighted an interesting use of deception where the defenders in a cyber-wargame experiment activated a CND tool to selectively redirect adversaries to a fake system that dynamically rewrote content from the real system. A shortcoming of the tool was that the adversaries gained access to both fake and real systems and could subsequently recognize some telltale signs of deception.

2.7 Topology Deception

Traceroute is the *de facto* probing technique and tool for topology mapping [31]. Traceroute deception is an example of internal defense described in [29] since the deception is done in the defender's network instead of the attacker's. For traceroute probes originating from a local network where latencies are typically low, the essential concern is to detect and craft responses to all types of probes whose source IP addresses indicate their origin from a network controlled by the defenders. Incoming probes from the wider Internet usually have to transit through various ASs and possibly crossing over into other geographical regions. Organizations deploying deception have to examine these additional aspects when designing their deception systems. They might also consider between two forms of traceroute deception, mainly inventing replies with spoofed IP addresses or manipulating replies based on the real physical topology. The first option offers the best flexibility at the cost of greater computing resources. The second option allows the network load to be balanced but constraints the deceptive topology due to the dependence on the physical network layout.

Prior work in this field [32] [33] detailed methodologies for manipulating the perception generated from an adversary's analysis of the traceroute probe responses. The implementation required modification of the router kernel in order to deliver custom fake responses to traceroute probes.

Trassare *et al.* [34] introduced a methodology for deceiving incoming traceroute probes and providing a configurable illusion of a network topology to the prober. Their effort concentrated on adversarial active probing, of which traceroute is the tool of choice.

Trassare *et al.* focus on benefits to the military counter-intelligence by providing both random and intelligent masking methods. The random method is akin to military radar jammers in an electronic warfare (EW) suite emitting false radio frequency (RF) signatures, and generates responses to traceroute probes with pseudorandom source addresses based on a random number generator. Although such deception may be unraveled by more sophisticated adversaries, the primary purpose is to add frustration and delay to the topology mapping process. The intelligent method is more elaborate and aims to turn the existing genuine topology into a deceptive version that shields critical infrastructure nodes of high value.

The implementation was performed by developing a Linux kernel module for a virtual Cisco 3725 router as a proof of concept. The experiment setup involved using the Graphical Network Simulator (GNS3) due to the ease of virtualization of hardware and software components. The custom kernel is an illustration of a deception system simulating false responses. Simulation helps to replace hardware decoys by taking on their specialized roles with respect to traceroute responses.

In order to unmask such deception, the prober has to adopt a multipronged approach of discovering inconsistencies in the deception by means of further probing with varied methods.

West built on on Trassare's work on detecting traceroute probes and developed a tool called DeTracer to present a fake network topology to incoming IPv4 probes [33]. The tool was able to portray the alternative routes for IP addresses of a given source-destination pair. It covered the usage of UDP, TCP, ICMP, paris-traceroute, scamper, and nmap traceroute probes and successfully responded with a deceptive route.

The initial experiment was carried on using virtual machines playing the role of a prober, DeTracer, and web server. Although there were some inconsistencies with the unrealistic probe RTTs, they disappeared when individual physical machines were deployed instead. In addition to the localized laboratory experiment, West deployed DeTracer on a single IP address on the public Internet. Global probers such as Ark would then be able to probe and obtain a fake traceroute result. The DeTracer demonstrated the traceroute replies containing made-up hops with IP addresses with realistic RTTs. It also replicated a portion of the fake path presented in the Pirate Bay case in Chapter 3.

2.8 Topology Deception Inconsistencies

This thesis assumes active deception is being implemented on some of the destination hosts, with the goal of intentionally deceiving incoming probes and portraying a divergent network topology. As such, the goal is to methodologically distinguish between honest and deceptive hosts. This fabricated topology may hide critical servers or show a honeypot disguised as a series of routers. We consider unusual but legitimate traceroutes that exhibit characteristics of an elaborate gag to be of interest. A known instance is the Christmas carol traceroute made viral in a reddit post [7]. It is possible that deception systems in the wild are not always perfect. Hence, an imperfect active deception system will exhibit inconsistencies and anomalies, which may indicate the use of deception in the return traceroute data. These imperfections would be most suitable for exploitation.

There is a particular field of note in every IPv4 packet called the IPv4 Identification (IPID) field. Its original intention was to provide a mechanism to distinguish fragments of one packet from another by having unique identifiers within the same source-destination pair and protocol [35]. Bellovin noted that consecutive packets originating from the same host would also contain IPID values of a sequential nature [36]. In theory, if a single host were to reply to traceroute probes on behalf of other hosts, the resultant replies would contain consecutive IPID values as well. More information on this field and its use as a possible exploitable inconsistency are covered in Chapter 3.

CHAPTER 3: Methodology

Our methodology is to obtain and analyze the CAIDA routed topology dataset by building a software parser that accepts scamper *warts* files and filters the set of traceroutes according to a set of predefined criteria. We use CAIDA traceroute data for 2013 and 2014, together with their corresponding DNS names from the CAIDA site and external AS datasets from the Route Views Project, as detailed in Section 3.2.5. The approach is to comb through entries in the 2013 dataset to identify traceroutes with certain interesting characteristics, filter and perform additional analysis on them, and compare the findings for equivalent entries in the 2014 dataset.

In addition, we utilize instances of known ground truth such as The Pirate Bay case and the Christmas carol themed traceroute mentioned briefly in Chapter 2 and reported on the Internet in recent years as case studies for classification and verification. Some of these instances of known ground truth are no longer operational, hence we must rely on analyzing historic traceroutes. In contrast, other examples of known topology deception were running at the time of this thesis work, hence we are able to perform additional probing and analysis. The inconsistencies found are compared against the actual deception strategy.

3.1 Types of Deception Inconsistencies

There are several possible discernible differences that may contribute to the unmasking of a deception system, but vary in their effectiveness. Certain differences are not definitive—an example of which will be timing delays that have many non-deceptive causes including load balancing and queues. In some cases, comparative results from multiple sources are stronger indicators, while others require a combination of results across multiple hops to be indicative. This section discusses the major types of deception inconsistencies that may be used as our filtering criteria. The following list is not exhaustive and acts as potential indicators of an anomaly, which may suggest either intentional or benign manipulation of the interred topology from traceroutes. It also serves as a starting point for our investigation into possible deception.

3.1.1 Packet Delay Correlation

Any deceit aiming to confuse the prober will have to respond with appropriate delays according to the adversary's deception strategy. The RTT, as explained in Section 2.5, is a loose metric for measuring delays. It provides only a lower bound on the feasible distance. The minimum delay also means that the deceiver will not be able to spoof responses shorter than what is possible from its physical geographical position. While the adversary could add delay via artificial means, to enforce the perception that the target is further away, it is not possible to decrease the delay significantly without a change of location.

Each traceroute hop can be checked for infeasible latencies based on their geolocated IP addresses and the prober's known location. For example, if the hop destination is known to be in a region 5,000 miles (8046.72km) from the prober, it will be suspicious to obtain a RTT of 40ms as it is physically unrealistic due to signal processing, delays from routers and switches, and so forth. Considering only propagation delay in a route composed of optic fiber, the observed delay of 40ms will mean a minimum distance of around 5,000 miles between the sender and receiver, as light in a vacuum travels at a reduced speed of approximately 66% [37] of the original speed of light. The lack of realism stems from the fact that the cumulative result from other types of delays (e.g, processing delays from routers along the way, etc.) will mean that the overall RTT will exceed the theoretical minimum.

We may also subject the target to additional TCP probing by sending a TCP SYN packet and observe the delay in its reply from the target. Under normal circumstances, the target replies with a TCP SYN/ACK packet, and the delay is recorded. If the target is running a deception operation for traceroutes, it is possible that the system may fail to manipulate the delay for ordinary TCP packets. For example, a traceroute may report a destination RTT of 600ms, while measurement of RTTs of the first two phases of the TCP three-way handshake yielded a consistent 100ms period. This could be due to injected latencies in the traceroute data. This technique can only be used against a live or operational host, unless such probing has been performed and recorded previously during the original traceroute.

3.1.2 Perceived Geographical Discrepancies

Resolved IP addresses using DNS, *whois*, or other IP geolocation techniques provide clues to the country in which the device is situated. If the countries identified in all hops are not adjacent and do not have an undersea link connecting them directly, this clue raises a possibility of a fake hop inserted into the traceroute response.

One instance of such a case is a traceroute report showing a hop from the continental United States to North Korea. News reports indicated that the entire North Korea's Internet traffic flowed through China Unicom routers, hence a direct link to the United States is not possible [38]. Another example could be when a traceroute report shows hops from continental United States to multiple countries in the Middle East and back to the United States in a successive manner. A packet is unlikely to traverse such a route unless it is on a deliberate circuit. A deliberate circuit may be a case of intentional manipulation or just plain misconfiguration.

3.1.3 Inconsistent Hop Limit Values

Each IPv4 packet contains a TTL or hop limit field, which acts a mechanism to control the lifetime of the packet as it traverses through the networks. In traceroutes, the TTL value is decremented by one for every hop the packet traverses. Hops are simply nodes, or more specifically in this case, routers in which the packet traverses en route to its destination, as discussed in Section 2.1. According to RFC 792 [12], the resultant ICMP Time Exceeded message sent has to include the first eight bytes of the original datagram's data. Therefore, responses from each router along the forward path should contain the original datagram's incrementing TTL values. This allows the prober to map ICMP responses to the packets it sent. The TTL value in the matched packet originally sent by the sender is the probe TTL. The quoted TTL is the based on the TTL value in the actual datagram received by the responding machine. Duplicate or inconsistent TTL values may indicate anomalies.

3.1.4 Continuous Consecutive IP Identification Values

As most operating systems implement an incrementing IPID for every packet they send out, as discussed in Section 2.8, and relying on the fact that it is unlikely for two hosts to have synchronized IPID velocities, it is unlikely that a list of resultant IPIDs from a traceroute will have sequential and monotonic IPID values. Subsequently, if there are consecutive IPIDs from two successive hops, it is potentially an anomaly and may indicate deception.

3.1.5 Autonomous System Link Discrepancies

An inconsistency in AS links is similar to the above-mentioned perceived geographical location discrepancy. Suppose two AS are known not to have a direct connection to each other and a traceroute indicates otherwise, this could be indicative of an anomaly. As AS links change from time to time, it is imperative that the date of BGP table data used to determine the AS links be as close to the traceroutes as possible.

3.1.6 Multiple Probe Types

An incomplete deception system may generate false traceroute returns for one type of probe and neglect another. Different types of probes were discussed in Section 2.1. Different responses from different probe type may indicate the presence of such deception. For instance, a deception system built specifically for deceiving UDP and ICMP traceroutes might have left out an equivalent implementation to respond to TCP probes. The prober can send multiple traceroutes based on UDP, TCP, and ICMP in an attempt to identify inconsistencies in RTT, TTL, IPID, and location, etc.

3.1.7 Multiple Ingress Points

Trassare noted that the deceptive topology presented to incoming probes should be consistent across all ingress points [32]. This is to say that traceroutes performed from multiple vantage points should infer network topologies that are not too divergent from one another. Therefore, the deception may fail if the prober discovered a significantly different topology when probing from another vantage point or had access to an internal machine that bypasses the deception mechanism completely.

3.1.8 Common Subnet Hops

The detection of multiple hops within the same subnet in a traceroute may indicate traversal through an AS or single organization network. As with the case with multiple ingress points, an incomplete or misconfigured deception topology may present cycles or loops in the traceroute traffic. For instance, if the traceroute shows a path taken through a subnet A, followed by an exit to some faraway location and then a return to subnet A, the ensuring loop may raise questions on the forwarding strategy.

3.2 Filtering the IPv4 Routed /24 Topology Dataset

The downloaded dataset [19] from CAIDA was cataloged according to team, year, cycle, and, finally, host monitor. Each team-year-cycle-host item is a compressed collection of traceroutes in Scamper's *warts* native binary format. On average, each *warts* file in 2013 contained 66,000 traceroutes. The 2013 data consists of 37,106 *warts* files grouped into 208 cycle folders. The cycle folders are a form of classification for IP addresses probed in a standard probing cycle of around 48 hours. Each cycle contains the individual output from around 17—18 geographically distributed unique monitors for each team, which are machines doing the actual probing work. The total disk space used for the 2013 /24 Topology Dataset was about 850 gigabytes. The 2014 dataset was slightly larger at about 900 gigabytes. Our main focus was on the 2013 dataset, with the 2014 dataset as a reference comparison. A leading motivation for this selective focus is the Pirate Bay case.

To scope the thesis, we elected to perform a limited set of filters as described in the subsequent subsections to sieve out interesting traces for more detailed examination. The consecutive IPIDs and duplicate TTLs filters were chosen due to their ease of implementation. The workflow is as follows:

1. Filter the 2013 dataset by consecutive IPIDs and duplicate TTLs.
2. Build a sorted summary of results.

3. Identify potential traceroutes for further analysis.
4. Filter selected traceroutes for faster DNS and AS matching.
5. Perform DNS and AS matching for each hop in the selected traceroutes.
6. Extract traceroutes in the 2014 dataset with the same destination as those previously identified in the 2013 dataset.
7. Perform DNS and AS matching on the extracted 2014 list.

3.2.1 Consecutive IPIDs

We scanned each *warts* file in 2013 for records that exhibit characteristics of continuous consecutive IPIDs. This filter algorithm parses the traceroute result and records the following information:

- *LCONSEC*. This is the largest number of hops containing a continuous consecutive IPIDs chain with the constraint that the detected chain must be at the end of the traceroute. As we are parsing historical data, the end of the traceroute will be based on the data that is available to us in the dataset. It may not be the actual destination due to the occurrence of non-responsive nodes. A chain can be understood as a series of hops in consecutive order. This constraint is based on the speculation that some deception systems spoof all the responses from a single network interface after a certain hop. An example can be seen in Table 3.1.
- *LDCONSEC*. This is similar to the above *LCONSEC* statistic, but it takes distinct IP addresses into consideration. Distinct addresses allow us to differentiate between seemingly erroneous chains of identical addresses and typical unique chains. An example can be seen in Table 3.2.
- *ACONSEC*. This is the largest number of hops containing a continuous consecutive IPIDs chain within the traceroute. It acts as a rough metric and does not cater for distinct addresses. It is also a looser form of the *LCONSEC* case in that chains do not have to terminate at the end of the traceroute. Table 3.3 demonstrates an instance of *ACONSEC*.
- *ADCONSEC*. This item is the fine-grained version of the above *ACONSEC* statistic that takes distinct IP addresses into consideration. Table 3.4 is an example of *ADCONSEC*.

IP Address	IPID
129.186.6.251	46325
129.186.254.131	36654
192.245.179.52	50752
146.57.253.138	1673
146.57.252.238	51149
206.108.255.4	16176
64.50.227.194	48640
216.165.136.34	3264
216.170.152.193	3265
216.170.152.193	3266
216.170.152.193	3267

Table 3.1: Example of *LCONSEC* chain with 3 continuous consecutive IPID hops ending at the last hop.

IP Address	IPID
66.163.76.145	36364
128.242.186.169	7511
63.146.26.229	0
208.46.163.207	7446
69.31.116.143	7447
69.31.33.1	15608
152.179.78.37	4816
152.63.17.61	7448
152.63.20.242	7449
152.63.19.57	7450
157.130.250.190	7451

Table 3.2: Example of *LDCONSEC* chain with 3 continuous consecutive IPID hops and distinct IP addresses ending at the last hop.

As traceroute results are not always perfect and contain missing responses, our parser will ignore the non-responsive hops, and continue to determine any consecutive IPID chains without them. The advantage of this implementation is that we can quickly identify consecutive chains. The downsides are that long chains may be broken into smaller chains, and very short chains of three hops may never be discovered. We leave the work of identifying potential chains with non-responsive hops to future work.

IP Address	IPID
188.1.50.9	24408
80.239.128.181	0
213.155.135.86	0
80.91.247.125	0
213.155.130.67	45234
192.205.37.49	6794
12.83.58.133	6975
70.238.215.3	6976
70.238.215.3	7977
151.164.102.11	13
95.76.129.140	988

Table 3.3: Example of *ACONSEC* chain with 3 continuous consecutive IPID hops without location constraints.

IP Address	IPID
4.69.153.6	24900
4.69.152.81	0
4.53.210.146	0
61.19.9.173	33482
61.19.7.129	19
61.19.7.246	0
61.19.15.106	21
61.91.213.129	22
61.91.213.38	23
61.91.213.180	24
119.46.147.54	45341

Table 3.4: Example of *ADCONSEC* chain with 3 continuous consecutive IPID hops with distinct IP addresses without location constraints.

3.2.2 Continuous Duplicate TTLs

We included a check for continuous duplicate TTL values and recorded the highest number of continuous duplicate TTL values for each traceroute. This check is to be performed together with the above consecutive IPID filtering such that the filtered results will either contain consecutive IPIDs and/or duplicate TTLs. We speculate that a high continuous duplicate TTL count may be a result of an incomplete deception system where the responding machine fakes new hops with IP address as the only variable component.

3.2.3 Filter Summaries

The next step after compiling the filtered traceroutes was to build a summary of the results for each year. The summary contains the various consecutive IPID and duplicate TTL statistics, as a function of the length of the chain. This summary allows us to quickly identify the number of traceroutes containing anomalous chains of different lengths. Because the number of anomalous chains decreases exponentially in their length, we obtain a manageable number of candidate traceroutes for more detailed examination.

3.2.4 Inconsistency Identification

Based on the filter summary, we identify a threshold point as our starting point of investigation. For instance, the traceroutes that fall within the top 10 highest number of consecutive IPIDs and duplicate TTLs may be considered as candidates for a second round of filtering. After that, we proceed with the DNS and AS matching process for a better picture of the perceived geographical locations of the IP addresses in each hop of the traceroutes.

3.2.5 DNS and AS Matching

The IPv4 Routed /24 DNS Names Dataset [22] had a different structure from the earlier topology dataset in that the data was grouped into year, month, and day. The DNS lookups were performed typically in a short time after each traceroute for better matching purposes. Each year-month-day item is a compressed collection of IP addresses and DNS names in plain text. In total, there were 805,523,249 IP-DNS entries in 2013. The 2013 dataset consumed about 48 gigabytes of disk space. The 2014 dataset had 874,568,595 entries and took up about 50 gigabytes of disk space.

We used the dataset from the University of Oregon Route Views Archive Project [23] to obtain the AS information. While the IPv4 Routed /24 AS Links Dataset [39] existed, it did not have direct IP address to AS mappings that we could use. The Route Views dataset was grouped by year-month directories and contained compressed binary files named according to the day and time they were obtained. We had to identify the required Routing Information Base (RIB) files based on the dates the candidate traceroutes were performed. This task was automated by having a script to extract the dates from the *warts* files containing our candidate traceroutes, and generating a list of RIB files for processing. Each routeviews RIB was processed with *bgpdump* [40] for our python script to start the matching process. Each compressed RIB file contained more than 10 million entries and averaged around 48 megabytes. The processed RIB file is in plain text form and contains details for a given prefix such as the AS path, origin, next hop, etc. We could then match an IP address in our traceroute hop to its corresponding prefix and obtain the AS associated with it.

3.2.6 Year-on-Year Comparison

We wanted to know if any of the identified traceroutes in the 2013 dataset exhibited the same behavior in 2014. Hence, a script was written to search for traceroutes in 2014 that matched the destination IP address of all identified traceroutes. The generated list was then sorted according to the 2013 dataset's order for easy side-by-side viewing. We continued with the DNS and AS matching to help us understand the results.

3.3 Case Studies of Traceroute Oddities

Here we look at a few case studies known to portray traceroute oddities. These case studies present us an opportunity to use the above inconsistencies as test criteria. Not all cases mean to deceive the prober on the actual network topology. Some of them are treated as gags, as their purpose is to spread a message compared to actual topology deception where the desire is to influence an adversary.

3.3.1 The Pirate Bay

Deception benefits from obscurity in that there is a lower likelihood of discovery. Publicizing a topology deception on the Internet tends to invite open sleuthing. The Pirate Bay was reported to be operating from North Korea [41] and traceroutes after the announcement confirmed that the responses ended with a North Korea IP address. The ensuing publicity brought more scrutiny on the deception as quick-responding Internet users posted their analyses to message boards [42] and blogs [43].

The Pirate Bay case is interesting as it showcases inconsistencies in the traceroute RTT when compared to a TCP three-way handshake's RTT. The true latency could be measured when a direct TCP connection was made to The Pirate Bay's web server [43]. If the traceroute RTT and the TCP three-way handshake's RTT were significantly different, it would be likely that the web server was actually in a different geographical location than previously reported by the traceroute RTT. This underscores the belief that traceroute deception can sometimes be unmasked by using non-traceroute probing methods. The other unmasking method used was to perform a traceroute to the spoofed IP address in North Korea and compare the routes [43]. Another intriguing point was the use of spoofed IP addresses which the deceiver did not own. It was in contrast to the other known ground truths such as the Christmas Carol in Section 4.4 that had some form of control over the IP addresses presented in their traceroute replies. The use of spoofed addresses offered greater flexibility but required more careful attention to mask the inconsistencies, as shown above.

We search for traceroutes pertaining to The Pirate Bay case (194.71.107.15) in the 2013 IPv4 Routed /24 Topology Dataset, as well as all IP addresses in its /24 prefix (194.71.107.*). We suspect that there is a chance of other hosted IP addresses in that subnet with similar form of deception. Next, we match the DNS records and AS paths for further inspection. Finally, the 2014 dataset may be mined to see if there are any similarities over time.

3.3.2 Christmas Carol Traceroute

The Christmas Carol traceroute was first publicized in reddit [7] as a holiday themed gag. It utilized reversed DNS names to IP addresses in the 82.133.91.0/25 subnet. Compared to the earlier Pirate Bay case, the IP addresses in the probe replies were real and not spoofed. Although DNS PTR names hint at an unofficial source, it was not known how the routing was accomplished. Since the gag is still operational, we perform a variety of traceroutes using different protocols to see if the gag topology holds. In addition, we probe the individual IP addresses in between and observe whether the route traveled is similar. Figure 3.1 shows a portion of the traceroute output symbolizing a Christmas tree.

```
17 ccr1009.futile.net [93.89.84.75]
18 ooooxooooooxooo.V.ooooooxooooxoooo [82.133.91.37]
19 ooxoooooxooooo.MMM.ooooooooxxoooxo [82.133.91.18]
20 oooxoooooxooo.EEEEE.oooxoooooxoooo [82.133.91.63]
21 ooooxooxooox.RRRRRRR.ooooooxooooox [82.133.91.56]
22 oxooooooxoo.RRRRRRRRR.oooxooooooxo [82.133.91.55]
23 xoooxooooo.YYYYYYYYYYY.oooxooooxoo [82.133.91.58]
24 ooxoooooxooooo.CCC.ooooooooxoooxoo [82.133.91.96]
25 oooooxooo.HHHHHHHHHHHHH.oxoooxoooo [82.133.91.23]
26 ooxooxoo.RRRRRRRRRRRRRRR.ooxoooxoo [82.133.91.49]
27 oxoooxo.IIIIIIIIIIIIIIIII.oooxooxo [82.133.91.60]
28 oooxoo.SSSSSSSSSSSSSSSSSSS.ooxoooo [82.133.91.42]
29 oooxoooxoooooo.TTT.ooooooooooooxoo [82.133.91.61]
30 ooxoo.MMMMMMMMMMMMMMMMMMMMMM.oooxo [82.133.91.34]
31 xxoo.AAAAAAAAAAAAAAAAAAAAAAAA.oxoo [82.133.91.80]
32 oxo.SSSSSSSSSSSSSSSSSSSSSSSSSS.ooo [82.133.91.40]
33 ooxooooooooooo.XXX.oooooooooooooxo [82.133.91.35]
34 oxoooooooooooo.XXX.ooooooooooooxxo [82.133.91.10]
35 Oh.the.weather.outside.is.frightful [82.133.91.41]
36 But.the.fire.is.so.delightful [82.133.91.19]
37 And.since.weve.no.place.to.go [82.133.91.77]
38 Let.It.Snow.Let.It.Snow.Let.It.Snow [82.133.91.43]
39 xXx [82.133.91.24]
```

Figure 3.1: Cropped traceroute output for xmas.futile.net.

3.3.3 Star Wars Traceroute

The Star Wars traceroute presented an opening crawl themed message crafted from reversed DNS names in an unused 206.214.251.0/24 subnet [3]. Unfortunately, Werber had terminated the experiment due to a denial of service attack soon after publicizing it. Figure 3.3 shows a portion of the traceroute output containing the opening crawl.

Werber's original setup included two Cisco 1841 Integrated Services Routers with VRF configured to bounce packets between each other. VRF lets the operator define multiple instances of a routing table on a router without introducing conflict due to IP address reuse [4], as referenced in Figure 3.3. He forwarded the packets at his border router (216.81.59.173) to one of the Cisco routers with VRF, which then forwards the packet to the other router. This was likely accomplished with static routes in each VRF. The two

```
10 Episode.IV [206.214.251.1]
11 A.NEW.HOPE [206.214.251.6]
12 It.is.a.period.of.civil.war [206.214.251.9]
13 Rebel.spaceships [206.214.251.14]
14 striking.from.a.hidden.base [206.214.251.17]
15 have.won.their.first.victory [206.214.251.22]
16 against.the.evil.Galactic.Empire [206.214.251.25]
17 During.the.battle [206.214.251.30]
18 Rebel.spies.managed [206.214.251.33]
19 to.steal.secret.plans [206.214.251.38]
20 to.the.Empires.ultimate.weapon [206.214.251.41]
21 the.DEATH.STAR [206.214.251.46]
22 an.armored.space.station [206.214.251.49]
23 with.enough.power.to [206.214.251.54]
24 destroy.an.entire.planet [206.214.251.57]
```

Figure 3.2: Cropped traceroute output for obiwan.scryne.net.

Cisco routers then pass the packets back and forth from each other until the end of the routing table has been reached. The length of the message shown is dependent on the length of the routing table on each router. By updating the DNS PTR records with separate parts of the Star Wars opening crawl, the reverse DNS lookups will show the complete message when all IP addresses have been matched to their respective DNS PTRs.

3.3.4 CMAND Experimental DeTracer

The DeTracer tool had the ability to spoof traceroute replies for UDP, TCP, ICMP, etc. [33], in order to counter multiple probe types. We are also aware of the DeTracer generating reasonably plausible RTTs. As such, we work on identifying other inconsistencies as detailed in section 3.1. The following are the potential inconsistencies that the DeTracer may have left out in its implementation:

1. Non-random sequence of IPIDs. As the DeTracer used a single machine and interface to respond to traceroute probes, it is possible that the reply IPIDs may be consecutive.
2. Fixed topology inferred from responses. If there are multiple ingress points for a probe to arrive at the DeTracer, and it was tasked to present a fixed network topology, probing results showing identical IP address chains after a certain hop could clue us

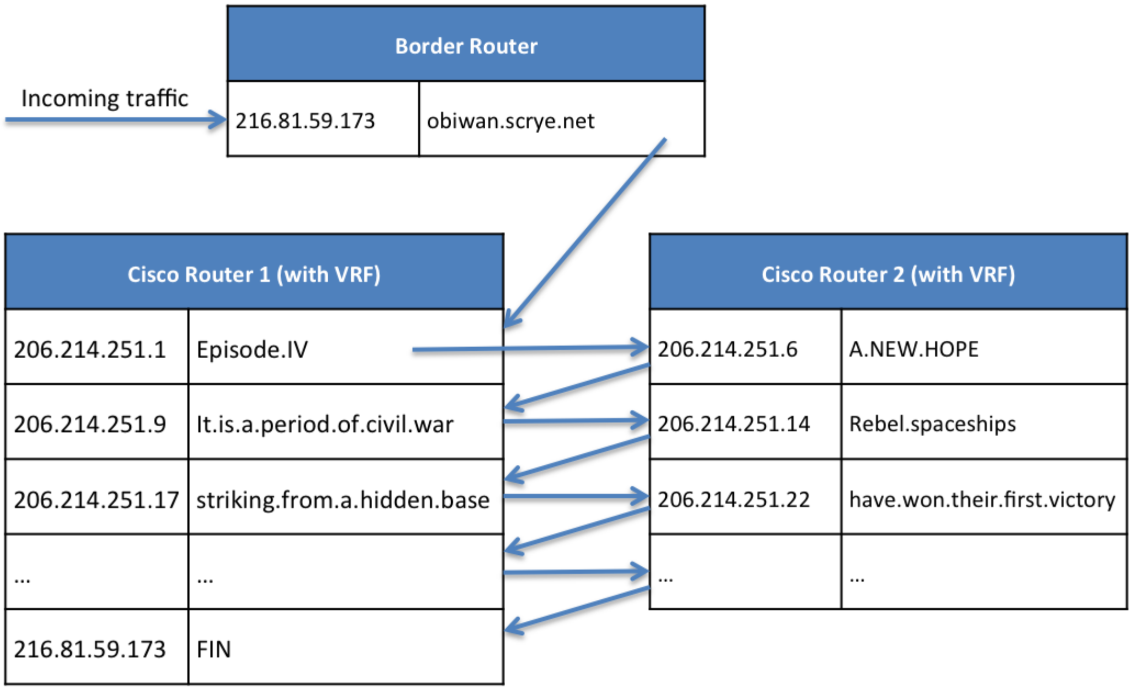


Figure 3.3: Approximate Star Wars gag setup.

on the possibility of deception.

3. Adversarial probing of individual IP addresses in the traceroute replies. We can attempt to probe each of the IP addresses in the traceroute response to gauge if they are really alive. A failure to respond would raise the likelihood of a spoofed IP address.

CHAPTER 4:
Results and Analysis

The goal of this thesis is to examine historic traceroute datasets for anomalous results indicative of potential deception, as well as to find evidence of deception by operational targets. Using the methodology described in Chapter 3, we hope to identify interesting anomalous patterns in an incremental manner. This chapter presents results from analyzing the IPv4 Routed /24 Topology Dataset and the individual case studies. We first present our findings on the 2013 dataset, followed by the 2014 dataset, and a comparison between findings in both years, before ending with an analysis of the case studies.

4.1 IPv4 Routed /24 Topology Dataset Findings

4.1.1 Findings in the 2013 Dataset

We generated a summary of the 2013 filtered dataset consisting of the consecutive IPID and duplicate TTL chains, grouped according to the chain lengths. Figure 4.1 shows the count of *LCONSEC* and *LDCONSEC*, inclusive of those traceroutes with 30 or fewer matching hops. The figure contains a log-based Y-axis due to the volume of traceroutes involved. (The complete summary, with all data and hops, can be found in the appendix.) There were *LCONSEC* chains with up to 699 hops, but they were one-off fringe cases. For instance, the chain with 699 hops had duplicated IP addresses for the hops that contained consecutive IPID values. There were also 64,508,982 traceroutes with at least two hops forming a single *LCONSEC* chain. These traceroutes consisted of 1.21% of the 5,313,384,228 traceroutes in the 2013 dataset. The majority of the *LCONSEC* chains were within 20 hops. 1,613 traceroutes had chains of 20 hops in length or fewer, and just three traceroutes had 21 hop chains. The blue and red bars in Figure 4.1 indicate *LCONSEC* and *LDCONSEC*, respectively. As the *LDCONSEC* count is a subset of *LCONSEC*, the number of traceroutes categorized as *LDCONSEC* is lower than that of LCONSEC.

Compared to *LCONSEC*, the *LDCONSEC* metric showed a drastically reduced set, with

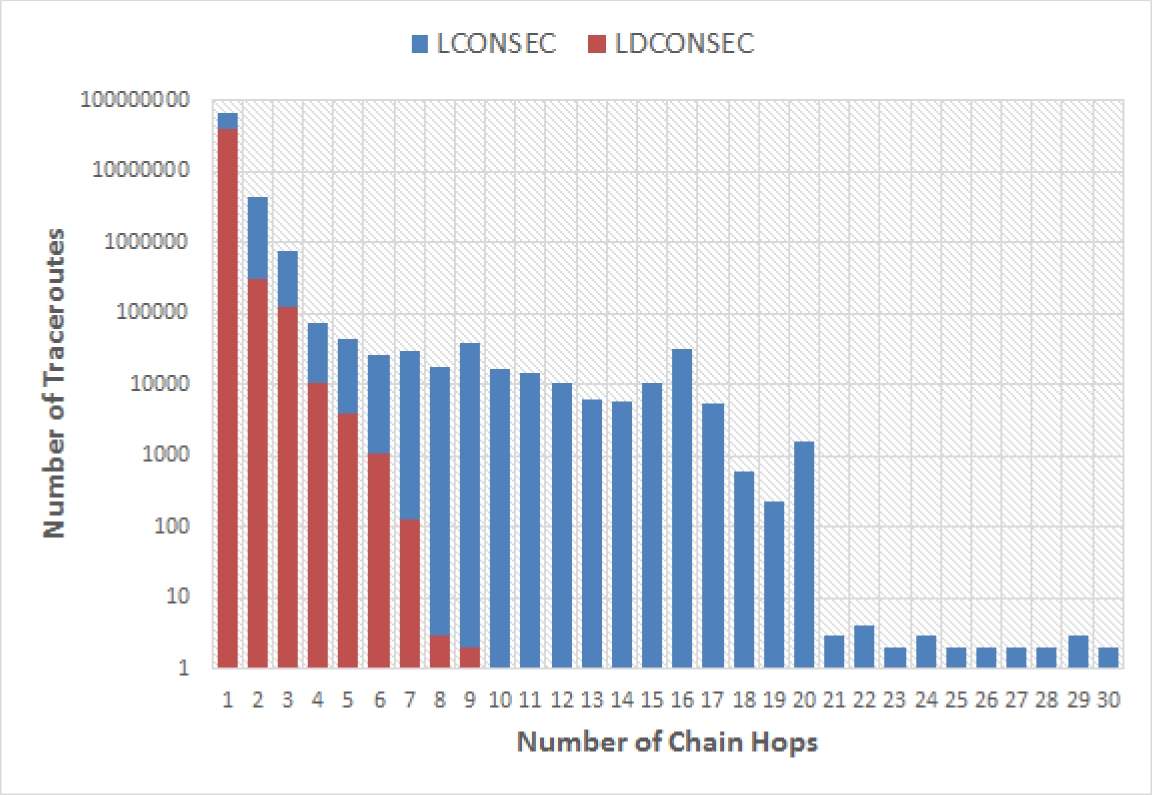


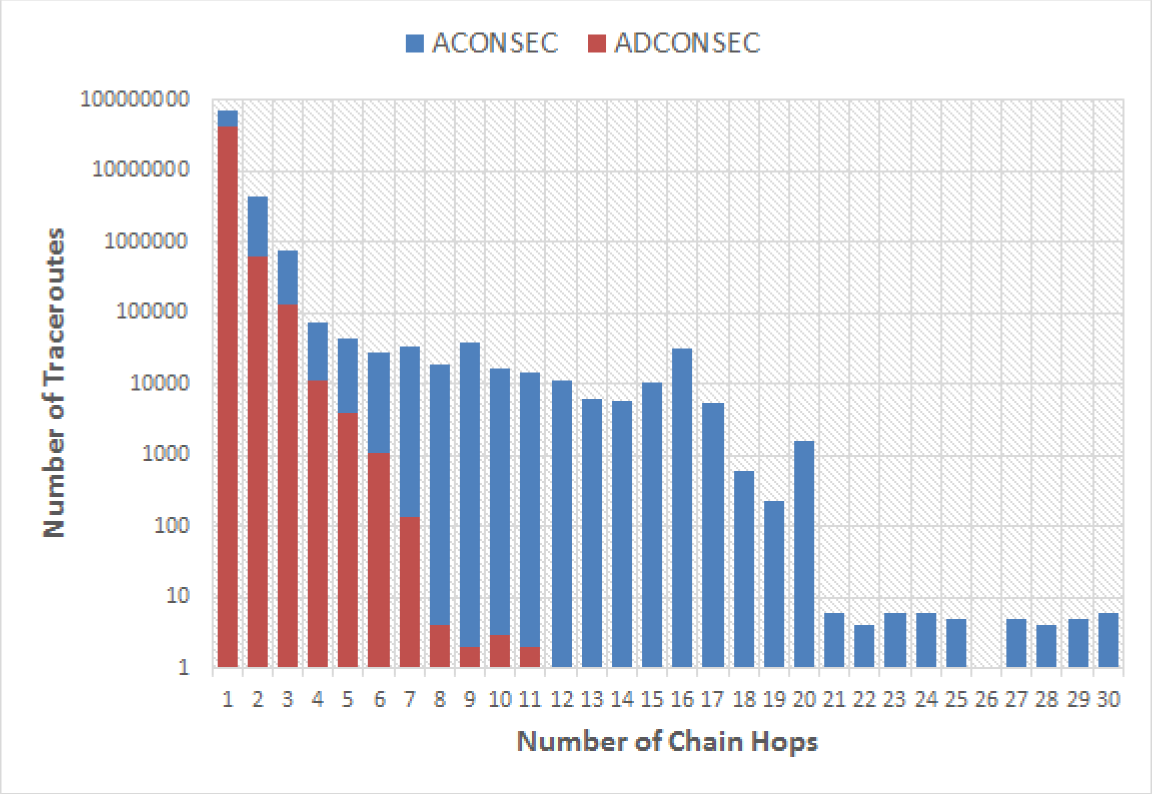
Figure 4.1: Summary of 2013 dataset containing *LCONSEC* and *LDCONSEC* chains of between 1 and 30 hops.

the curve falling off to the low digits after seven hops. There were 39,336,467 (56%) *LDCONSEC* out of 70,026,420 *LCONSEC* instances. There were only three traceroutes with eight *LDCONSEC* hop chains against 131 traceroutes with seven hop chains. This result meant that a significant number of traceroutes had consecutive IPID chains with duplicate IP addresses. Table 4.1 shows an example of a traceroute with hops containing duplicate IP addresses. The traceroute had a consecutive IPID chain of 2,363 hops which caused it to be listed under the *ACONSEC* category. As the IPID chain was not detected at the end of the traceroute, it was not listed under the *LCONSEC* category. There were 4,066 out of 4,077 hops with the IP address `118.155.197.13` in the traceroute, and their reply TTLs always had a value of 245. The ellipsis represented the sections of the traceroute missing as the IP address, probe TTL, and reply TTL did not change. For instance, the hop after the first hop with `118.155.197.13`, IPID of 16006, probe TTL of 14 and reply TTL of 245, had the same `118.155.197.13`, probe and reply TTLs as before, but with a consecutive

addr	ipid	probettl	replyttl	rtt
134.197.113.14	2909	1	64	330
134.197.0.33	58371	2	63	325
207.197.33.69	62988	3	62	6633
137.164.26.21	62922	4	61	3504
137.164.25.46	16939	5	60	12903
137.164.25.33	47986	6	59	17572
137.164.26.134	0	7	249	17578
198.71.45.20	0	8	248	50067
208.100.127.34	58550	9	247	55366
208.100.120.26	15989	12	245	61394
198.62.88.194	16005	14	245	63815
118.155.197.13	16006	14	245	65241
...	...	...	...	...
118.155.197.13	16126	14	245	338371
118.155.197.13	16129	14	245	340390
118.155.197.13	16125	15	245	67025
118.155.197.13	16127	15	245	70111
...	...	...	...	...
118.155.197.13	20073	15	245	4715190
118.155.197.13	20074	15	245	4717636
118.155.197.13	20075	16	245	67086

Table 4.1: Example of a traceroute to `76.165.200.63` on 12 October 2013 (daily.l7.t2.c002780.20131012.rno-us.warts.gz) containing a long chain of 2,363 hops. Note the duplicate IP addresses (`118.155.197.13`) and TTLs (245) in its consecutive IPID chain (16005-20075).

IPID of 16007. This continued until the `118.155.197.13` hop with an IPID of 16126, after which there was a gap in the consecutive numbering. The hops with `118.155.197.13` generally followed a consecutive IPID pattern but the probe TTL value grew from 14 to 15. There were breaks in the IPID pattern which resulted in a shortened consecutive IPID count. The last hop ended with a probe TTL of 16. With the probe TTL field being only eight bits long and could only contain a maximum value of 255, it was impossible to obtain a traceroute response of 4,077 hops. Hence, the anomaly here would likely to be a result of malfunctioning responses instead of actual deception. This example highlights traceroutes with high consecutive IPID count without distinct IP addresses. Duplicate IP addresses might have been caused by load balancing on the responding side as their TTL values are identical. It is also possible that router misconfiguration as well as zero TTL forwarding could lead to the duplication of IP addresses. We noted the ascending RTTs values for each of the probe TTL, which seemed to reset after a transition to the next probe TTL. Although scamper uses the Paris traceroute to enhance the probing precision, the results are not always without flaws due to the possibility of random per-packet load balancing, as noted by its authors [13]. Unless this historic traceroute phenomenon is still active and reproducible, it is difficult to find out more information on the possible causes.



.

Figure 4.2: Summary of 2013 dataset containing *ACONSEC* and *ADCONSEC* chains of between 1 and 30 hops.

The *ACONSEC* and *ADCONSEC* metrics show that the number of traceroutes with ACONSEC and ADCONSEC chains of length up to seven hops is proportional to that of the *LCONSEC* and *LDCONSEC* results, as seen in Figure 4.2. These two measurements are supersets of the earlier *LCONSEC* and *LDCONSEC* measurements. They do not distinguish between continuous chains occurring at the end of the traceroute and elsewhere. We observe that at a count of seven hops, there is only a difference of nine traceroutes between

LDCONSEC (131 traceroutes) and *ADCONSEC* (141 traceroutes). This observation indicates that the bulk of the continuous consecutive IPID chains transpired at the end of the responding traceroute. It also steers us towards using only the *ADCONSEC* output in our next step of analysis.

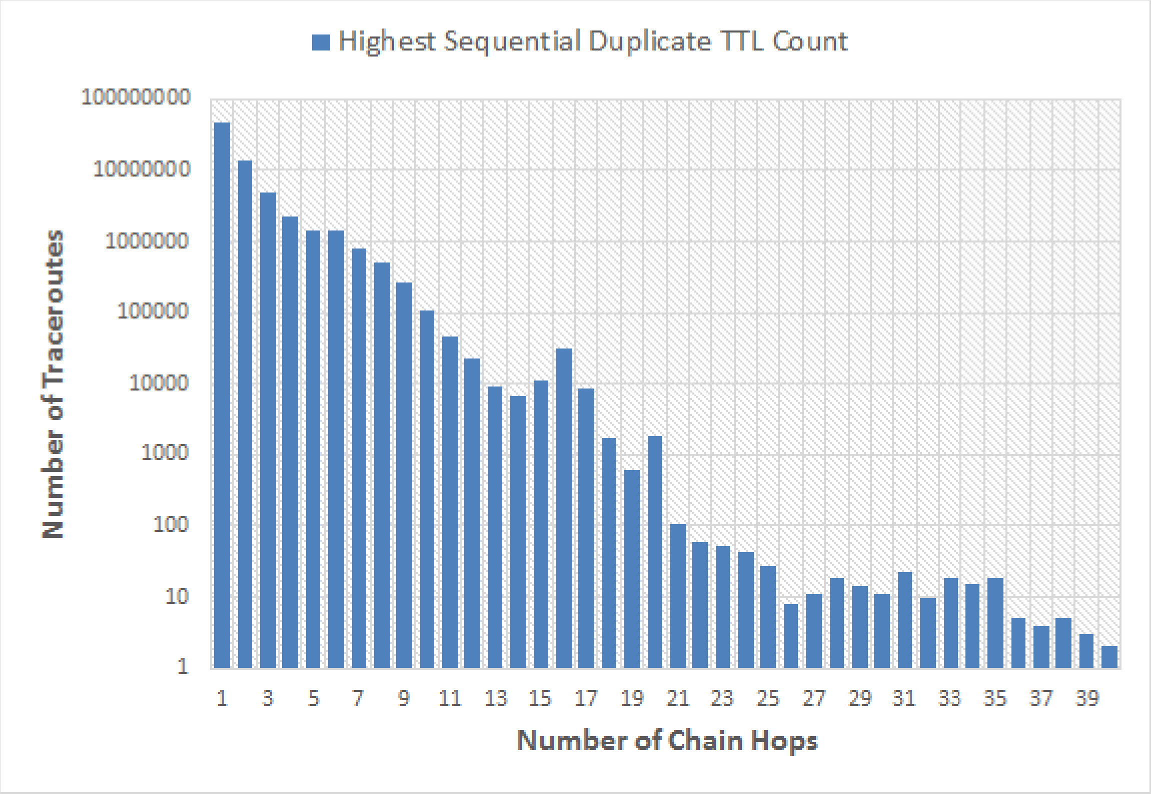


Figure 4.3: Summary of 2013 dataset containing the largest number of sequential duplicate TTLs chains (between one and 40 hops)

Figure 4.3 presents the findings on the number of traceroutes containing the largest number of continuous duplicate TTLs chains of between 1 and 40 hops. We observe a general downward trend, and there were no large dips in the number of duplicate TTL chains after a certain number of hops.

4.1.2 Pattern Identification in the 2013 Dataset

Our initial methodology was to only filter the traceroutes with continuous consecutive IPIDs. However, we noticed that several traceroutes had identical or looping hops with the same IP address. We then modified the filter to incorporate a distinct IP address checker, which produces the *LDCONSEC* and *ADCONSEC* results. In doing so, we were able to narrow down the number of traceroutes where consecutive IPIDs were more likely to be indicative of deception. We selected traceroutes with an ADCONSEC chain count of at least seven hops to be extracted and analyzed for any major patterns. The choice of seven hops was based on the summary result, to find a chain length that would produce a small enough set of traceroutes for more detailed manual analysis. Including only those chains of length seven or greater reduced the candidate traceroutes to a manageable set size (whereas six or greater produced thousands of candidates). By selecting chains of seven hops and above, we are left with 151 traceroutes to analyze. We leave the work of discovering whether the excluded traceroutes (those outside of the ADCONSEC filter) contain interesting artifacts to future work.

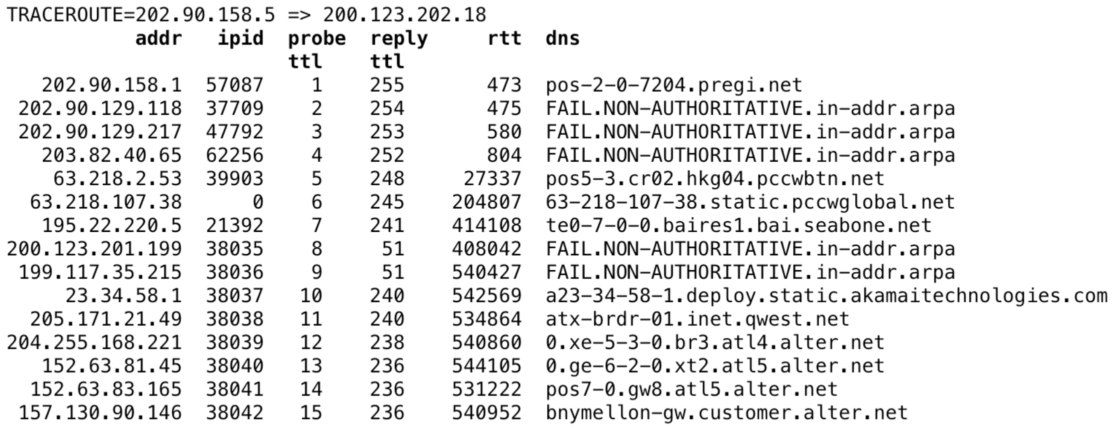
As we had identified instances of traceroutes with duplicate reply TTLs earlier, our next course of action was to determine if the duplicate reply TTLs were of use in complimenting our 151 *ADCONSEC* traceroutes. For traceroutes with duplicate reply TTLs, we observed that some of them contain duplicate probe TTLs, as well. The reply TTL is the TTL in the ICMP packet sent by the responding machine. As there were 71,520 cases of duplicate TTL chains of seven hops and above, and the *ADCONSEC* chains tended to fall in number at around seven hops, we decided, in the interest of time, to leave the task of correlating these duplicate TTL chains to the possibility of deception to future work.

The 151 traceroutes were identified across 60 unique monitors in three teams with 62% of all monitors reporting three potentially deceptive traceroutes and below. While the median was two traceroutes per monitor, the highest number of reported traceroutes from a single monitor came from the Beijing monitor at 11 traceroutes. There were eight destination IP addresses belonging to at least two traceroutes and one of them was the destination for three traceroutes. While the individual source and destination IP addresses could be reused, every traceroute had a unique pair of source and destination IP addresses. With these 151 traceroutes identified as containing continuous consecutive IPIDs chains of at least seven hops, we proceeded to perform the tasks of DNS and AS path matching. For the AS paths, there was an additional path check which ascertained if there was a link from one hop to the next by identifying the pair of corresponding AS entries for both hops, and combing through the associated RIB file from the same time period as the traceroute to find the AS pair in the list of all AS paths. Section 3.1.5 describes the exploitation of AS link discrepancies to provide an additional metric for discovering traceroutes with possible deception.

4.1.3 Major Patterns in 2013

We discovered a recurring pattern in at least 79 out of the 151 filtered traceroutes. Since the size of the TTL field is a single byte, the range of possible values a TTL field can hold is from 0 to 255. Default TTLs are configurable, and the presence of large jumps in the reply TTL values may indicate different hardware or operating systems. These traceroutes had a consecutive IPID chain that began with the first two hops using only seven out of eight bits in their reply TTLs before reverting back to the full eight-bit reply TTLs as before the appearance of the chain. This could be a natural consequence of the default TTL values for the machine's operating system. The third hop in the chain typically resolved to a DNS name (PTR record) like `deploy.static.akamaitechnologies.com` belonging to Akamai Technologies, Inc, as seen in Figure 4.4. Another resolved DNS name contained `airtel.in`, which belongs to Akamai Technologies India Pvt Ltd, as it was reported in a *whois* lookup. In Figure 4.4, we expected the hop with `63.218.2.53` to have a reply TTL of 251 as the previous hop had a reply TTL of 252. This would hint at a router with different default TTL or that there were hidden routers. We observed that the traceroute did not reach the destination as the last responding hop was not the destination address. A common accompanying pattern was that the last responding hop usually resolved to a name containing `customer.alter.net`. We found that only eight out of 90 traceroutes with a resolved last hop to a `customer.alter.net` address did not contain a hop with an Akamai related name. A *whois* lookup at this time of writing showed that the `alter.net` domain belongs to Verizon Business Global LLC. In addition, there were around five hops after the hop with the Akamai address. We call this pattern the *Akamai* pattern for an easy reference.

Under normal circumstances, the quoted TTL of each hop of an ICMP Time Exceeded reply in traceroute responses would be a value of one. A router probed by traceroute would typically decrement the incoming probe TTL by one. If the decremented TTL value was

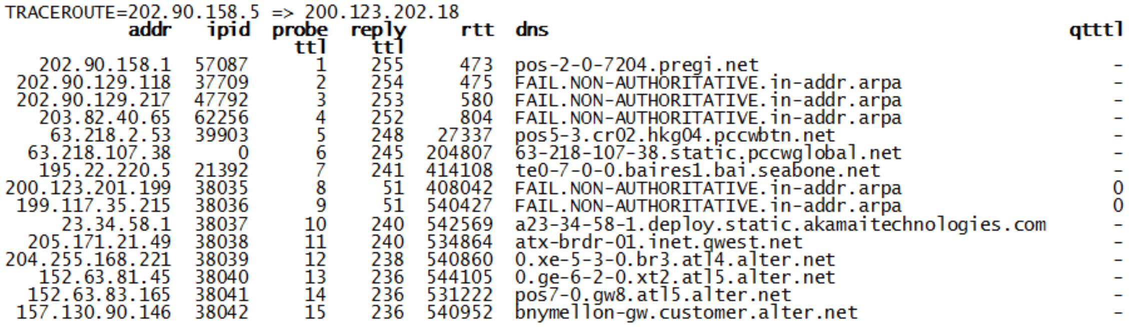


```
TRACEROUTE=202.90.158.5 => 200.123.202.18
           addr   ipid  probe  reply      rtt  dns
                        ttl    ttl
    202.90.158.1  57087    1    255       473  pos-2-0-7204.pregi.net
  202.90.129.118  37709    2    254       475  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  202.90.129.217  47792    3    253       580  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    203.82.40.65  62256    4    252       804  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     63.218.2.53  39903    5    248     27337  pos5-3.cr02.hkg04.pccwbtn.net
   63.218.107.38      0    6    245    204807  63-218-107-38.static.pccwglobal.net
    195.22.220.5  21392    7    241    414108  te0-7-0-0.baires1.bai.seabone.net
 200.123.201.199  38035    8     51    408042  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  199.117.35.215  38036    9     51    540427  FAIL.NON-AUTHORITATIVE.in-addr.arpa
      23.34.58.1  38037   10    240    542569  a23-34-58-1.deploy.static.akamaitechnologies.com
   205.171.21.49  38038   11    240    534864  atx-brdr-01.inet.qwest.net
 204.255.168.221  38039   12    238    540860  0.xe-5-3-0.br3.atl4.alter.net
    152.63.81.45  38040   13    236    544105  0.ge-6-2-0.xt2.atl5.alter.net
   152.63.83.165  38041   14    236    531222  pos7-0.gw8.atl5.alter.net
  157.130.90.146  38042   15    236    540952  bnymellon-gw.customer.alter.net
```

Figure 4.4: An Akamai pattern traceroute to 200.123.202.18 on 18 April 2013 (daily.l7.t1.c002479.20130418.mnl-ph.warts.gz) with a hop containing a DNS name related to Akamai Technologies.

zero, the router would prevent the packet from being forwarded and reply with an ICMP Time Exceeded datagram instead. Hence, the incoming probe TTL would have to be a value of one for the packet to be discarded and the ICMP datagram sent as a response. We discovered the first two hops in the consecutive IPID chain of all traceroutes with the Akamai pattern had quoted TTLs with values of zero. Augustin *et al.* described a traceroute anomaly that if the quoted TTL value of a hop was zero, it could mean the previous router had a misconfiguration that resulted in packets with a TTL value of zero being forwarded to the current receiving router [13]. This would suggest that the hop preceding the consecutive IPID chain in the traceroutes with the Akamai pattern might have forwarded the packet to the next router even after decrementing the probe TTL to zero, thus causing the responding hop to have a quoted TTL of zero.

Multiprotocol Label Switching (MPLS) provides a method to speed up router forwarding in fixed paths by avoiding IP header analysis for every packet [44]. The MPLS architecture introduces a Label Stack Entry (LSE) header which contains information on the forwarding decisions to be made by MPLS routers. An IP router forwarding MPLS traffic through MPLS routers would insert the LSE header before the IP header in the packet to be forwarded. The LSE header contains its own TTL field. Upon receiving a packet, the MPLS router will decrement the TTL value in the LSE header and not the TTL field in the IP header, unless the `ttl-propagate` option is active. The `ttl-propagate` option in most



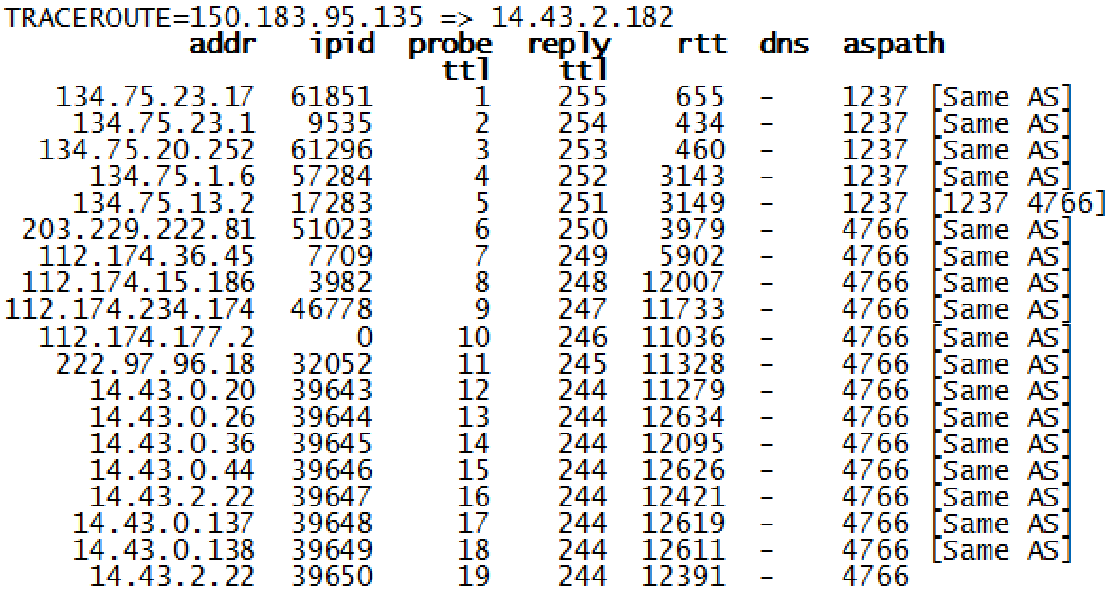
```
TRACEROUTE=202.90.158.5 => 200.123.202.18
           addr   ipid  probe  reply      rtt  dns                                          qtttl
                          ttl    ttl
    202.90.158.1  57087      1    255      473  pos-2-0-7204.pregi.net                          -
  202.90.129.118  37709      2    254      475  FAIL.NON-AUTHORITATIVE.in-addr.arpa             -
  202.90.129.217  47792      3    253      580  FAIL.NON-AUTHORITATIVE.in-addr.arpa             -
    203.82.40.65  62256      4    252      804  FAIL.NON-AUTHORITATIVE.in-addr.arpa             -
     63.218.2.53  39903      5    248    27337  pos5-3.cr02.hkg04.pccwbtn.net                   -
   63.218.107.38      0      6    245   204807  63-218-107-38.static.pccwglobal.net             -
    195.22.220.5  21392      7    241   414108  te0-7-0-0.baires1.bai.seabone.net               -
 200.123.201.199  38035      8     51   408042  FAIL.NON-AUTHORITATIVE.in-addr.arpa             0
  199.117.35.215  38036      9     51   540427  FAIL.NON-AUTHORITATIVE.in-addr.arpa             0
      23.34.58.1  38037     10    240   542569  a23-34-58-1.deploy.static.akamaitechnologies.com -
   205.171.21.49  38038     11    240   534864  atx-brdr-01.inet.qwest.net                      -
 204.255.168.221  38039     12    238   540860  0.xe-5-3-0.br3.atl4.alter.net                   -
    152.63.81.45  38040     13    236   544105  0.ge-6-2-0.xt2.atl5.alter.net                   -
   152.63.83.165  38041     14    236   531222  pos7-0.gw8.atl5.alter.net                       -
  157.130.90.146  38042     15    236   540952  bnymellon-gw.customer.alter.net                 -
```

Figure 4.5: An Akamai pattern traceroute to 200.123.202.18 on 18 April 2013 (daily.l7.t1.c002479.20130418.mnl-ph.warts.gz) showing hops with a quoted TTL value of zero.

routers ensures that the TTL in the original IP packet will be copied to the TTL field in the LSE header. This allows MPLS routers to respond with ICMP Time Exceeded replies to traceroute probes. Donnet *et al.* showed that MPLS tunnels with the `ttl-propagate` option enabled in their MPLS routers would still result in hops with quoted TTLs of one [45]. Therefore, a typical traceroute would contain hops with quoted TTL values of one. The quoted TTL could have a value of more than one if the traceroute probe terminates at a responding MPLS router with `ttl-propagate` enabled, and replies with an ICMP Time Exceeded datagram containing the quoted TTL value. The effect is that quoted TTLs above a value of one are likely indicative of the number of MPLS routers in the route. The presence of the quoted TTLs of values zero and more than one may not suggest real deception. They may possibly be merely artifacts of MPLS routing or router misconfiguration and incorrect implementation. We leave this identification of the causes to future work.

A variation of the above Akamai pattern involved a third hop with an unresolved DNS name or belonging to ISPs such as `seed.net.tw`, `merca.net.co`, etc. At times, the IPID chain would end after the first two hops with the two-digit TTL values, only to resume the sequence soon after. These mixed cases comprised of approximately 10% of the filtered traceroutes.

The second recurring pattern had 14 cases of consecutive IPID chains with IP addresses regularly within the `14.43.0.*` and `14.43.2.*` range. Figure 4.6 showed the last and second last hops before the consecutive IPID chain were always non-zero and zero IPID values, respectively. The reply TTLs in the chain were mostly identical, with an outlier traceroute

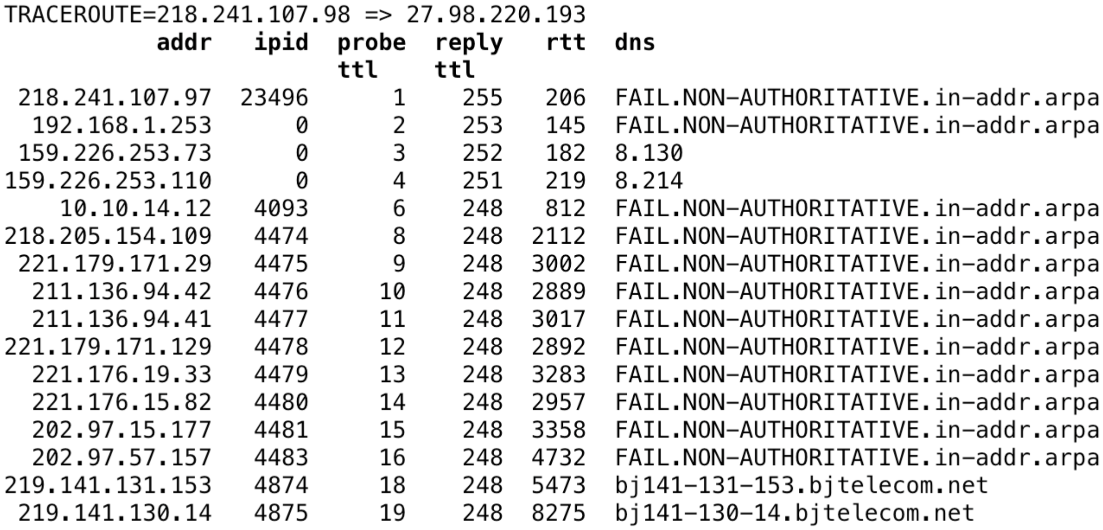


```
TRACEROUTE=150.183.95.135 => 14.43.2.182
              addr    ipid  probe  reply     rtt  dns   aspath
                             ttl    ttl
      134.75.23.17   61851     1    255      655   -    1237 [Same AS]
       134.75.23.1    9535     2    254      434   -    1237 [Same AS]
     134.75.20.252   61296     3    253      460   -    1237 [Same AS]
        134.75.1.6   57284     4    252     3143   -    1237 [Same AS]
       134.75.13.2   17283     5    251     3149   -    1237 [1237 4766]
    203.229.222.81   51023     6    250     3979   -    4766 [Same AS]
     112.174.36.45    7709     7    249     5902   -    4766 [Same AS]
    112.174.15.186    3982     8    248    12007   -    4766 [Same AS]
   112.174.234.174   46778     9    247    11733   -    4766 [Same AS]
     112.174.177.2       0    10    246    11036   -    4766 [Same AS]
      222.97.96.18   32052    11    245    11328   -    4766 [Same AS]
        14.43.0.20   39643    12    244    11279   -    4766 [Same AS]
        14.43.0.26   39644    13    244    12634   -    4766 [Same AS]
        14.43.0.36   39645    14    244    12095   -    4766 [Same AS]
        14.43.0.44   39646    15    244    12626   -    4766 [Same AS]
        14.43.2.22   39647    16    244    12421   -    4766 [Same AS]
       14.43.0.137   39648    17    244    12619   -    4766 [Same AS]
       14.43.0.138   39649    18    244    12611   -    4766 [Same AS]
        14.43.2.22   39650    19    244    12391   -    4766
```

Figure 4.6: A Korean Telecom pattern traceroute to 14.43.2.108 on 26 January 2013 (daily.l7.t1.c002346.20130126.cjj-kr.warts.gz) with IPID chains with IP ranges in 14.43.0.* and 14.43.2.*.

having multiple duplicated TTLs in this chain. The trend for 10 of the 14 cases was to have the first hop of the consecutive IPID chain to be 14.43.0.18, followed by 14.43.0.28, 14.43.0.34, the sixth hop being 14.43.0.137, and the seventh hop being 14.43.0.138. The chain length tended to be eight hops long. This little pattern would vary slightly in the later 2014 dataset, as detailed in Section 4.1.4. Although we were unable to resolve their DNS names, a current *whois* lookup revealed that the 14.32.0.0—14.95.255.255 range belonged to Korea Telecom. Hence, we identified this trend as the *Korean Telecom* pattern.

While the Korean Telecom pattern had the probes traversing through the IP ranges in 14.43.0.* and 14.43.2.*, another variation of this pattern used IP addresses based in China. We performed a *whois* lookup and found that the addresses belonged to Chinese companies such as China Telecom and China Mobile. All 13 traceroutes with this pattern continued to have duplicate TTL values in their IPID chains. An interesting subpattern was the appearance of a private address in the 10.10.0.0/16 subnet in four cases as evident in Figure 4.7. We therefore termed this pattern as the *China Telecom* pattern.

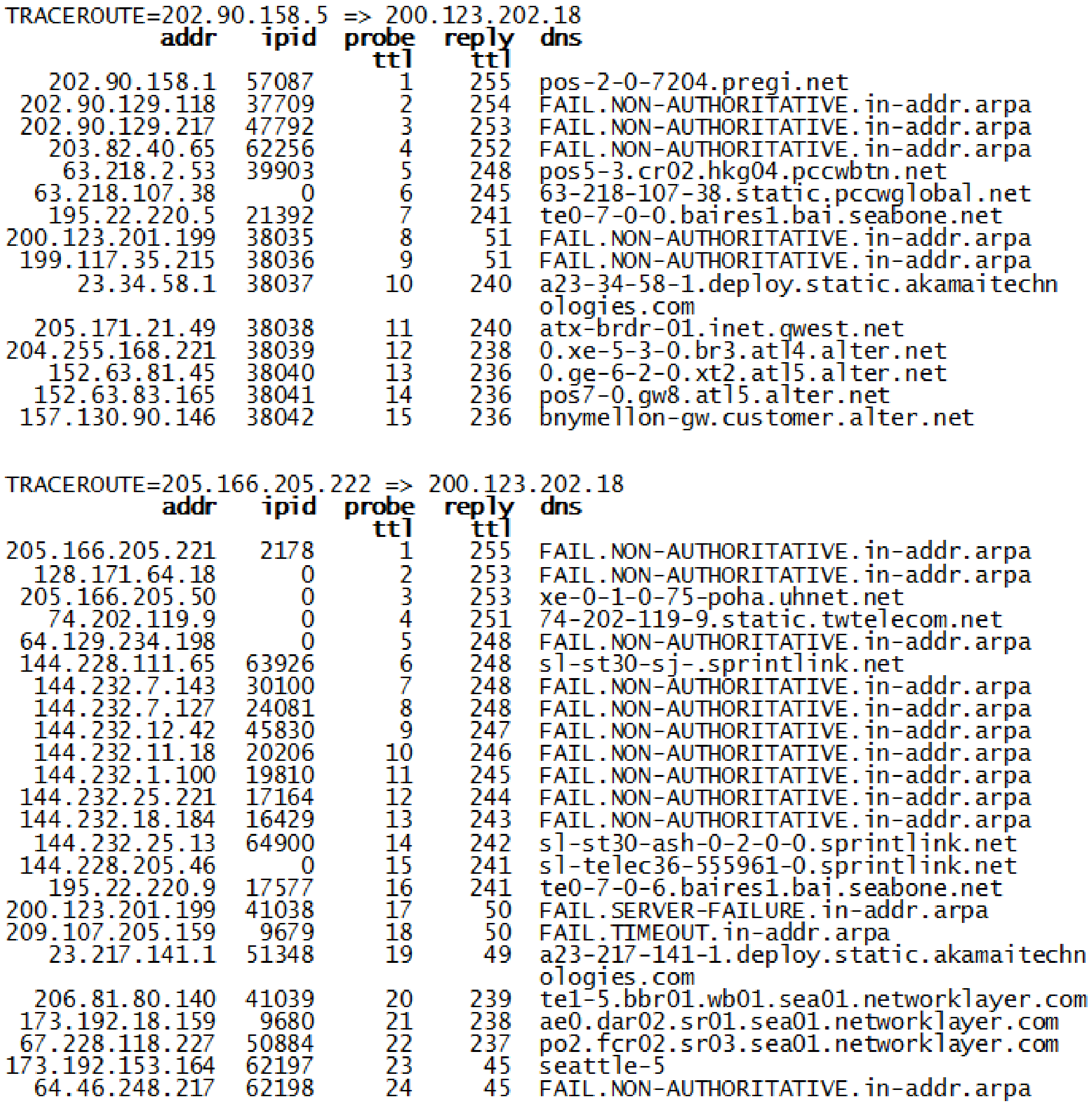


```
TRACEROUTE=218.241.107.98 => 27.98.220.193
           addr   ipid  probe  reply   rtt  dns
                        ttl    ttl
  218.241.107.97  23496      1    255   206  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   192.168.1.253      0      2    253   145  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  159.226.253.73      0      3    252   182  8.130
 159.226.253.110      0      4    251   219  8.214
     10.10.14.12   4093      6    248   812  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 218.205.154.109   4474      8    248  2112  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  221.179.171.29   4475      9    248  3002  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   211.136.94.42   4476     10    248  2889  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   211.136.94.41   4477     11    248  3017  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 221.179.171.129   4478     12    248  2892  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   221.176.19.33   4479     13    248  3283  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   221.176.15.82   4480     14    248  2957  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   202.97.15.177   4481     15    248  3358  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   202.97.57.157   4483     16    248  4732  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 219.141.131.153   4874     18    248  5473  bj141-131-153.bjtelecom.net
  219.141.130.14   4875     19    248  8275  bj141-130-14.bjtelecom.net
```

Figure 4.7: A China Telecompattern traceroute to 27.98.220.193 on 7 January 2013 (daily.l7.t2.c002316.20130107.pek-cn.warts.gz) with IPID chains with IP addresses assigned to China.

4.1.4 Looking Ahead in 2014

We matched the identified traceroutes in 2013 against their equivalents in the 2014 dataset. The matching was done by comparing the destination IP address and not the source IP address. The source IP address are different due to the random selection of destination addresses by each team in an Ark probing cycle, as detailed in Section 2.4. Figure 4.8 highlights an example in the 2014 dataset for our identified traceroutes in the 2013 dataset. The reply TTL value drop was still present but the long consecutive IPID chains were nowhere to be found. We observed that some of the destination addresses belonging to the Akamai pattern in the 2013 dataset had their `*.alter.net` hops replaced with hops of different IP address that resolved to `*.level3.net`. There was substantial variation in the routing topology of the filtered traceroutes over the 2014 period as we extracted disparate routes from traceroute destinations with multiple probing dates. In one instance, a traceroute from `119.40.82.245` to `157.238.81.44` on 9 March 2014 had its last few responding hops with a `*.cincinnati1.level3.net` address, while another traceroute from `200.160.7.159` to the same `157.238.81.44` on 13 May 2014 had different IP addresses resolved to `*.alter.net` instead. Additionally, we noted the continuity of the drop in reply TTL values despite the changes in the IP addresses.



```
TRACEROUTE=202.90.158.5 => 200.123.202.18
            addr    ipid  probe  reply  dns
                           ttl    ttl
    202.90.158.1   57087     1    255   pos-2-0-7204.pregi.net
  202.90.129.118   37709     2    254   FAIL.NON-AUTHORITATIVE.in-addr.arpa
  202.90.129.217   47792     3    253   FAIL.NON-AUTHORITATIVE.in-addr.arpa
    203.82.40.65   62256     4    252   FAIL.NON-AUTHORITATIVE.in-addr.arpa
     63.218.2.53   39903     5    248   pos5-3.cr02.hkg04.pccwbtn.net
   63.218.107.38       0     6    245   63-218-107-38.static.pccwglobal.net
    195.22.220.5   21392     7    241   te0-7-0-0.baires1.bai.seabone.net
 200.123.201.199   38035     8     51   FAIL.NON-AUTHORITATIVE.in-addr.arpa
  199.117.35.215   38036     9     51   FAIL.NON-AUTHORITATIVE.in-addr.arpa
      23.34.58.1   38037    10    240   a23-34-58-1.deploy.static.akamaitechn
                                        ologies.com
   205.171.21.49   38038    11    240   atx-brdr-01.inet.qwest.net
 204.255.168.221   38039    12    238   0.xe-5-3-0.br3.atl4.alter.net
    152.63.81.45   38040    13    236   0.ge-6-2-0.xt2.atl5.alter.net
   152.63.83.165   38041    14    236   pos7-0.gw8.atl5.alter.net
  157.130.90.146   38042    15    236   bnymellon-gw.customer.alter.net

TRACEROUTE=205.166.205.222 => 200.123.202.18
            addr    ipid  probe  reply  dns
                           ttl    ttl
 205.166.205.221    2178     1    255   FAIL.NON-AUTHORITATIVE.in-addr.arpa
   128.171.64.18       0     2    253   FAIL.NON-AUTHORITATIVE.in-addr.arpa
  205.166.205.50       0     3    253   xe-0-1-0-75-poha.uhnet.net
    74.202.119.9       0     4    251   74-202-119-9.static.twtelecom.net
  64.129.234.198       0     5    248   FAIL.NON-AUTHORITATIVE.in-addr.arpa
  144.228.111.65   63926     6    248   sl-st30-sj-.sprintlink.net
   144.232.7.143   30100     7    248   FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.7.127   24081     8    248   FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.12.42   45830     9    247   FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.11.18   20206    10    246   FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.1.100   19810    11    245   FAIL.NON-AUTHORITATIVE.in-addr.arpa
  144.232.25.221   17164    12    244   FAIL.NON-AUTHORITATIVE.in-addr.arpa
  144.232.18.184   16429    13    243   FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.25.13   64900    14    242   sl-st30-ash-0-2-0-0.sprintlink.net
  144.228.205.46       0    15    241   sl-telec36-555961-0.sprintlink.net
    195.22.220.9   17577    16    241   te0-7-0-6.baires1.bai.seabone.net
 200.123.201.199   41038    17     50   FAIL.SERVER-FAILURE.in-addr.arpa
 209.107.205.159    9679    18     50   FAIL.TIMEOUT.in-addr.arpa
    23.217.141.1   51348    19     49   a23-217-141-1.deploy.static.akamaitechn
                                        ologies.com
   206.81.80.140   41039    20    239   te1-5.bbr01.wb01.sea01.networklayer.com
  173.192.18.159    9680    21    238   ae0.dar02.sr01.sea01.networklayer.com
  67.228.118.227   50884    22    237   po2.fcr02.sr03.sea01.networklayer.com
 173.192.153.164   62197    23     45   seattle-5
   64.46.248.217   62198    24     45   FAIL.NON-AUTHORITATIVE.in-addr.arpa
```

Figure 4.8: Traceroute to 200.123.202.18 on 20 January 2014 (daily.l7.t2.c002955.20140120.hnl-us.warts) in the 2014 dataset, compared with the earlier traceroute in 2013, showing a modified topology and fragmented IPID values.

For the Korean Telecom pattern, there was also increased fragmentation in the IPID values such that there were fewer long continuous consecutive IPID chains. These newer traceroutes retained their duplicate TTL values in their IPID chains. However, most chains were merely sequential, having hops that skipped an incremented IPID value, as shown in Fig-

```
TRACEROUTE=150.183.95.135 => 14.43.2.108
          addr   ipid  probe  reply  dns
                         ttl    ttl
   134.75.23.17  61713      1    255  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    134.75.23.1  44552      2    254  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  134.75.20.252   3790      3    253  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     134.75.1.6  20974      4    252  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    134.75.13.2  49648      5    251  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 128.134.40.249  52792      6    250  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  112.174.36.17  46147      7    249  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 112.174.15.190  45527      8    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
112.174.234.174  34285      9    247  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  112.174.177.2      0     10    246  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   222.97.96.18  58409     11    245  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.0.18  61468     12    244  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.0.28  61469     13    244  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.0.34  61470     14    244  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.0.42  61471     15    244  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.2.22  61472     16    244  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    14.43.0.137  61473     17    244  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    14.43.0.138  61474     18    244  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.2.22  61475     19    244  FAIL.NON-AUTHORITATIVE.in-addr.arpa
```

```
TRACEROUTE=192.87.102.98 => 14.43.2.108
          addr   ipid  probe  reply  dns
                         ttl    ttl
  192.87.102.97  31969      1    255  vl163.sw14.amsterdam1.surf.net
 145.145.19.169      0      2    254  ae3.1131.jnr01.asd001a.surf.net
 195.69.145.150  59894      3     61  30gigabitethernet1-3.core1.ams1.he.net
   72.52.92.213  38334      4     61  100ge9-1.core1.lon2.he.net
   72.52.92.166  19597      5     60  100ge1-1.core1.nyc4.he.net
   72.52.92.226   6714      6     59  10ge10-3.core1.lax1.he.net
184.105.246.198      0      7    249  koreatelecom.10gigabitethernet12-7.
                                      core1.lax1.he.net
 112.174.87.209      0      8    249  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 112.174.83.117      0      9    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  112.174.8.121  44530     10    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 112.174.15.242  15644     11    247  FAIL.NON-AUTHORITATIVE.in-addr.arpa
112.174.235.174  14575     12    246  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  112.174.177.2      0     13    245  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   222.97.96.18  17037     14    244  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.0.20  18249     15    243  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.0.28  18250     16    243  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.0.36  18253     17    243  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.0.42  18255     18    243  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     14.43.2.22  18256     19    243  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    14.43.0.137  18258     20    243  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    14.43.0.138  18259     21    243  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   77.67.67.138  18260     22    243  art-gw.ip4.gtt.net
```

Figure 4.9: Traceroute to `14.43.2.108` on 8 March 2014 (daily.l7.t2.c003039.20140308.ams-nl.warts.gz) in the 2014 dataset, compared with the earlier traceroute in 2013.

ure 4.9. The `14.43.0.18` pattern first observed in Section 4.1.3 continued to appear with a slight variation. While the first, sixth, and seventh hops contained the same IP addresses as before, the second and third hops could have different IP addresses. The maximum number of hops for the sequential IPID chain was still eight hops.

The China Telecom pattern experienced substantial changes in that the IPID chains and duplicate TTLs were no longer seen in traceroutes with hops containing the `bjtelecom.net` domain. Figure 4.10 represents the general changes in the 2014 dataset. While the route in the later traceroute was different, it did not exhibit a large series of hops with duplicate TTLs and its IPID values were not consecutive.

```
TRACEROUTE=218.241.107.98 => 27.98.220.193
           addr   ipid  probe  reply  dns
                          ttl    ttl
 218.241.107.97  23496      1    255  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  192.168.1.253      0      2    253  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 159.226.253.73      0      3    252  8.130
159.226.253.110      0      4    251  8.214
    10.10.14.12   4093      6    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
218.205.154.109   4474      8    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 221.179.171.29   4475      9    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  211.136.94.42   4476     10    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  211.136.94.41   4477     11    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
221.179.171.129   4478     12    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  221.176.19.33   4479     13    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  221.176.15.82   4480     14    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  202.97.15.177   4481     15    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  202.97.57.157   4483     16    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
219.141.131.153   4874     18    248  bj141-131-153.bjtelecom.net
 219.141.130.14   4875     19    248  bj141-130-14.bjtelecom.net

TRACEROUTE=205.166.205.222 => 27.98.220.193
           addr   ipid  probe  reply  dns
                          ttl    ttl
205.166.205.221  61665      1    255  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  128.171.64.18      0      2    253  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   74.202.119.9      0      3    252  74-202-119-9.static.twtelecom.net
  66.192.249.22      0      4    249  sjc1-pr1-xe-0-3-0-0.us.twtelecom.net
  64.132.69.106  12958      5    249  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   202.97.90.73  23965      6    249  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  202.97.52.249  19444      7    249  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  202.97.53.241  54685      8    248  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   202.97.53.41  36986      9    247  FAIL.NON-AUTHORITATIVE.in-addr.arpa
219.141.162.174    621     10    246  bj141-162-174.bjtelecom.net
 219.141.130.18  47986     11    245  bj141-130-18.bjtelecom.net
```

Figure 4.10: Traceroute to 27.98.220.193 on 5 January 2014 (daily.l7.t2.c002926.20140105.hnl-us.warts.gz) in the 2014 datasets, compared with the earlier traceroute in 2013.

4.1.5 Limitations

A limitation of our continuous consecutive IPID filter was that continuous sequential but not consecutive chains were left out after the initial round of filtering. These could be long chains that had hops which did not have IPID values that conformed to the desired pattern. The hops might be exhibiting IPID values that incremented in small steps of greater than one, or had a few hops with duplicate IPID values as the previous hop. Some of the stepped values could be attributed to non-responsive hops, causing a small jump in the IPID value.

4.2 The Pirate Bay and Company

The original Pirate Bay destination IP address published in the reports on traceroute deception was `194.71.107.15` [41]. The blogger Will, noted on 5th March 2013, that his traceroute from his machine at `84.116.229.73` to `194.71.107.15` was ending with `202.72.96.6` in Phenom Penh, Cambodia, followed by `175.45.177.217` in Pyongyang, North Korea [9]. In our CAIDA 2013 dataset, we found a traceroute from `200.160.7.159` to `194.71.107.15` on 15th May 2013, and while the last responding hop was `175.45.177.217`, the second last connecting hop was instead `219.158.32.174`, as shown in Figure 4.11. A *whois* lookup shows that IP address `219.158.32.174` currently points to a location in China. Both traceroutes converge at the hop with the common IP address `213.198.77.122` and diverge as the subsequent IP addresses differ. They have the same `175.45.177.217` in their last hops. The IP addresses in between the hops with `213.198.77.122` and `175.45.177.217` were unique and not reused by each other.

The CAIDA traceroute displayed a generally monotonically increasing series of RTT values, which was expected with typical traceroutes. The reply TTLs had irregularities such as duplicate, missing, and rising reply TTLs. The hops containing `129.250.9.50` and `217.149.32.206` had the same reply TTL value of 241. The reply TTL for the hop with `217.149.32.206` was 241, but the next hop (`213.206.128.184`) had a reply TTL value of 243. There was also a case of a skipped TTL between the hops of `129.250.5.174` with a TTL of 247 and `129.250.2.64` with a TTL of 245.

The hop with IP address `213.198.77.122` had a markedly different base reply TTL from

```
TRACEROUTE=200.160.7.159 => 194.71.107.15
           addr   ipid  probe  reply     rtt  dns
                         ttl    ttl
  200.160.7.252      0      1    255     208  ae1-11.ar4.nu.registro.br
  200.160.0.253      0      2    254    1325  ae0-0.core1.nu.registro.br
  200.160.0.163      0      3    253   23446  xe-5-0-0-0.core1.gc.registro.br
  200.160.0.175      0      4    252    1419  xe-0-0-0-0.gw1.gc.registro.br
   159.63.48.37      0      5    251    1661  ge-2-0-1.ar4.gru1.gblx.net
   67.16.132.10  25570      6    248  170223  po1-20g.ar2.mia2.gblx.net
   67.16.132.10  25712      7    248  185586  po1-20g.ar2.mia2.gblx.net
  129.250.9.117  56356      8    247  130710  xe-0-0-0-10.r05.miamfl02.us.bb.gin.
                                              ntt.net
  129.250.2.184  56305      9    248  119777  ae-4.r20.miamfl02.us.bb.gin.ntt.net
   129.250.2.99  16357     10    246  160994  ae-8.r21.asbnva02.us.bb.gin.ntt.net
  129.250.2.145   2796     11    244  253019  ae-2.r23.amstnl02.nl.bb.gin.ntt.net
  129.250.2.159  61266     12    241  235343  ae-2.r02.amstnl02.nl.bb.gin.ntt.net
   129.250.2.65  33216     13    244  211509  xe-4-1.r02.dsdfge01.de.bb.gin.ntt.net
  129.250.5.173  18466     14    247  236615  xe-3-4.r00.dsdfge02.de.bb.gin.ntt.net
 213.198.77.122   9328     15     57  213572  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  129.250.5.174  32801     17    247  238479  xe-3-2.r02.dsdfge01.de.bb.gin.ntt.net
   129.250.2.64   2443     18    245  254946  xe-0-1-0-20.r02.amstnl02.nl.bb.gin.
                                              ntt.net
   129.250.9.50  19774     19    241  242506  xe-0.sprint.amstnl02.nl.bb.gin.ntt.net
 217.149.32.206   1007     20    241  244541  sl-bb21-ams-0-0-0.sprintlink.net
213.206.129.143  11726     21    243  256363  sl-bb23-lon-0-4-0-0.sprintlink.net
213.206.128.184  62431     22    242  258374  sl-crs2-lon-0-8-0-0.sprintlink.net
  144.232.9.163  61842     23    241  318459  sl-crs1-nyc-0-5-2-0.sprintlink.net
  144.232.5.216  14702     24    240  338347  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  144.232.18.59  20290     25    239  338393  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.1.73  42899     26    238  344416  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  144.232.11.17  24182     27    237  382504  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  144.232.12.41  19465     28    236  362183  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  144.232.7.124  43185     29    235  365264  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 144.232.18.106  23270     30    234  393365  sl-st20-sj-0-0-0.sprintlink.net
 144.223.242.82   8874     31    233  613360  sl-china6-192107-0.sprintlink.net
 219.158.32.174  31587     32    232  700001  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 175.45.177.217  59715     33    231  693447  FAIL.NON-AUTHORITATIVE.in-addr.arpa
```

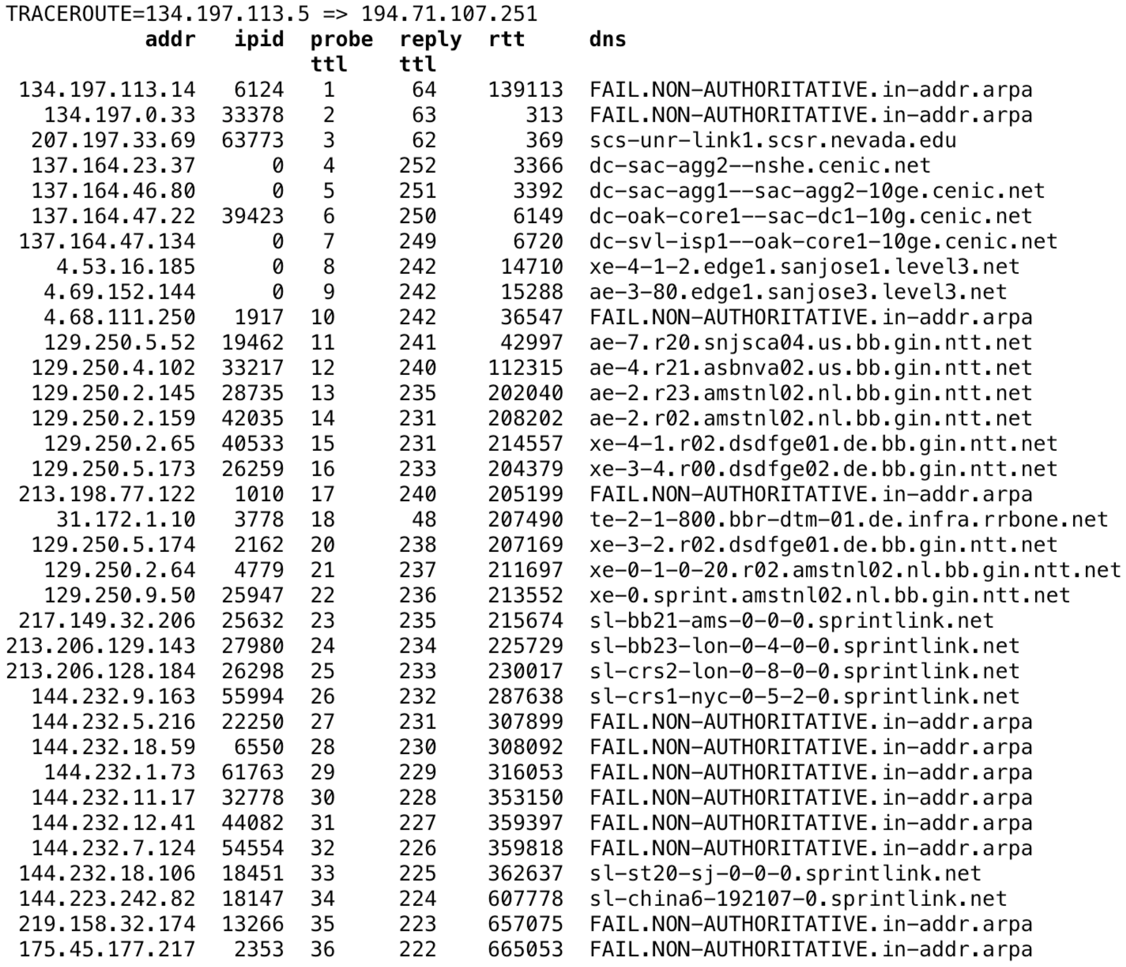
Figure 4.11: Traceroute to The Pirate Bay (194.71.107.15) on 15 May 2013 (daily.l7.t1.c002522.20130515.sao-br.warts.gz) in the 2013 dataset.

the rest of the hops. It had a seven-bit value compared to the eight-bit values in the rest of the hops, which was suggestive of either a router misconfiguration or more likely a different operating system. Moreover, the path after 213.198.77.122 had been modified such that the IP addresses were different, but in general, they were inferring a route through the United States to North Korea, as per the route in Will's blog from Germany to the United States, next to Cambodia, and lastly to North Korea [9]. One plausible reason would be that the deceptive topology had been modified to erase the inconsistency that there could not be any direct link between Cambodia and North Korea at that time. According to Will, a real AS link from Cambodia to North Korea did not exist [9]. We found that the AS links

```
TRACEROUTE=200.160.7.159 => 194.71.107.15
          addr   ipid  probe  reply     rtt  aspath
                         ttl    ttl
  200.160.7.252      0      1    255     208  22548 [Same AS]
  200.160.0.253      0      2    254    1325  22548 [Same AS]
  200.160.0.163      0      3    253   23446  22548 [Same AS]
  200.160.0.175      0      4    252    1419  22548 [22548 3549]
   159.63.48.37      0      5    251    1661  3549 [Same AS]
   67.16.132.10  25570      6    248  170223  3549 [Same AS]
   67.16.132.10  25712      7    248  185586  3549 [3549 2914]
  129.250.9.117  56356      8    247  130710  2914 [Same AS]
  129.250.2.184  56305      9    248  119777  2914 [Same AS]
   129.250.2.99  16357     10    246  160994  2914 [Same AS]
  129.250.2.145   2796     11    244  253019  2914 [Same AS]
  129.250.2.159  61266     12    241  235343  2914 [Same AS]
   129.250.2.65  33216     13    244  211509  2914 [Same AS]
  129.250.5.173  18466     14    247  236615  2914 [Same AS]
 213.198.77.122   9328     15     57  213572  2914 [Same AS]
  129.250.5.174  32801     17    247  238479  2914 [Same AS]
   129.250.2.64   2443     18    245  254946  2914 [Same AS]
   129.250.9.50  19774     19    241  242506  2914 [2914 1239]
 217.149.32.206   1007     20    241  244541  1239 [Same AS]
213.206.129.143  11726     21    243  256363  1239 [Same AS]
213.206.128.184  62431     22    242  258374  1239 [Same AS]
  144.232.9.163  61842     23    241  318459  1239 [Same AS]
  144.232.5.216  14702     24    240  338347  1239 [Same AS]
  144.232.18.59  20290     25    239  338393  1239 [Same AS]
   144.232.1.73  42899     26    238  344416  1239 [Same AS]
  144.232.11.17  24182     27    237  382504  1239 [Same AS]
  144.232.12.41  19465     28    236  362183  1239 [Same AS]
  144.232.7.124  43185     29    235  365264  1239 [Same AS]
 144.232.18.106  23270     30    234  393365  1239 [Same AS]
 144.223.242.82   8874     31    233  613360  1239 [1239 4837]
 219.158.32.174  31587     32    232  700001  4837 [4837 131279]
 175.45.177.217  59715     33    231  693447  131279
```

Figure 4.12: Traceroute to The Pirate Bay (194.71.107.15) with matching AS path data and links in the 2013 dataset.

between each hop in the CAIDA traceroute were feasible and connected according to the historic RIB data in Figure 4.12. The difference was that the Cambodian IP address was replaced by a Chinese IP address which belonged to a valid upstream AS for the North Korea IP address. This possibility could be further augmented by the fact that there was a time period of at least a month between the traceroute, suggesting Will's findings discussing a Cambodian link conducted on 4 March 2013, and the CAIDA traceroute conducted later on 15 May 2013. This would have provided an opportunity to tweak the deception to be more convincing. Our later search in the 194.71.107.0/24 subnet space discovered that traceroutes prior to 5 March 2013 did not have ending hops with 175.45.177.217 as evidenced in our subsequent findings. The CAIDA traceroute to 194.71.107.15 had little or no obvious inconsistencies such as duplicate TTLs and consecutive IPIDs. Subsequent analysis in this section shows that this IP played a major role in the topology deception. We were unable to obtain a resolved DNS name for 213.198.77.122. A whois lookup shows that 213.198.77.122 is under NTT America.



```
TRACEROUTE=134.197.113.5 => 194.71.107.251
          addr   ipid  probe  reply  rtt      dns
                       ttl    ttl
  134.197.113.14   6124   1     64   139113  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    134.197.0.33  33378   2     63      313  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   207.197.33.69  63773   3     62      369  scs-unr-link1.scsr.nevada.edu
   137.164.23.37      0   4    252     3366  dc-sac-agg2--nshe.cenic.net
   137.164.46.80      0   5    251     3392  dc-sac-agg1--sac-agg2-10ge.cenic.net
   137.164.47.22  39423   6    250     6149  dc-oak-core1--sac-dc1-10g.cenic.net
  137.164.47.134      0   7    249     6720  dc-svl-isp1--oak-core1-10ge.cenic.net
     4.53.16.185      0   8    242    14710  xe-4-1-2.edge1.sanjose1.level3.net
    4.69.152.144      0   9    242    15288  ae-3-80.edge1.sanjose3.level3.net
    4.68.111.250   1917  10    242    36547  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    129.250.5.52  19462  11    241    42997  ae-7.r20.snjsca04.us.bb.gin.ntt.net
   129.250.4.102  33217  12    240   112315  ae-4.r21.asbnva02.us.bb.gin.ntt.net
   129.250.2.145  28735  13    235   202040  ae-2.r23.amstnl02.nl.bb.gin.ntt.net
   129.250.2.159  42035  14    231   208202  ae-2.r02.amstnl02.nl.bb.gin.ntt.net
    129.250.2.65  40533  15    231   214557  xe-4-1.r02.dsdfge01.de.bb.gin.ntt.net
   129.250.5.173  26259  16    233   204379  xe-3-4.r00.dsdfge02.de.bb.gin.ntt.net
  213.198.77.122   1010  17    240   205199  FAIL.NON-AUTHORITATIVE.in-addr.arpa
     31.172.1.10   3778  18     48   207490  te-2-1-800.bbr-dtm-01.de.infra.rrbone.net
   129.250.5.174   2162  20    238   207169  xe-3-2.r02.dsdfge01.de.bb.gin.ntt.net
    129.250.2.64   4779  21    237   211697  xe-0-1-0-20.r02.amstnl02.nl.bb.gin.ntt.net
    129.250.9.50  25947  22    236   213552  xe-0.sprint.amstnl02.nl.bb.gin.ntt.net
  217.149.32.206  25632  23    235   215674  sl-bb21-ams-0-0-0.sprintlink.net
213.206.129.143  27980  24    234   225729  sl-bb23-lon-0-4-0-0.sprintlink.net
213.206.128.184  26298  25    233   230017  sl-crs2-lon-0-8-0-0.sprintlink.net
   144.232.9.163  55994  26    232   287638  sl-crs1-nyc-0-5-2-0.sprintlink.net
   144.232.5.216  22250  27    231   307899  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.18.59   6550  28    230   308092  FAIL.NON-AUTHORITATIVE.in-addr.arpa
    144.232.1.73  61763  29    229   316053  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.11.17  32778  30    228   353150  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.12.41  44082  31    227   359397  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   144.232.7.124  54554  32    226   359818  FAIL.NON-AUTHORITATIVE.in-addr.arpa
  144.232.18.106  18451  33    225   362637  sl-st20-sj-0-0-0.sprintlink.net
  144.223.242.82  18147  34    224   607778  sl-china6-192107-0.sprintlink.net
  219.158.32.174  13266  35    223   657075  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   175.45.177.217   2353  36    222   665053  FAIL.NON-AUTHORITATIVE.in-addr.arpa
```

Figure 4.13: Traceroute to 194.71.107.251 on 24 November 2013 (daily.l7.t2.c002850.20131124.rno-us.warts.gz) in the 2013 Dataset.

An exhaustive search of CAIDA traceroutes in the 194.71.107.0/24 subnet was performed using the 2013 dataset. We discovered that the topology deception was more extensive than previously thought. There were 128 cases of traceroutes that contained a hop with the 213.198.77.122 IP address and subsequent series of hops were either 17 or 18 hops long. The phenomenon was observed in traceroutes to the 194.71.107.0/24 subnet in our 2013 and 2014 datasets from 5 March 2013 to 10 December 2014 (645 days). The reported traceroute dates with the phenomenon meant that the deception had been active for at least 1 year, 9 months, and 5 days. For instance, a traceroute from 129.119.99.169 to 194.71.107.82 on 1 October 2014 had the series of hops with 213.198.77.122, while another traceroute from 216.66.30.102 to 194.71.107.82 on 20 December 2014 no longer exhibited this phenomenon. This was consistent with the date that The Pirate Bay was supposedly taken down in a police raid on its servers in Stockholm on 9 December 2014 [46]. Interestingly, the later traceroutes to 194.71.107.15 on both 2 February and 22 February 2014 did not have any deception in place. These series of hops had identical IP addresses except that the series with 18 hops had an additional hop at 31.172.1.10 in between 213.198.77.122 and the common 17-hop series. There were 110 instances of 17-hop series and 18 instances of 18-hop series. Figure 4.13 presents an 18-hop series that includes 31.172.1.10. In a typical 17-hop series, the hop with 213.198.77.122 exhibited a low seven-bit reply TTL while in the 18-hop series, hops with 31.172.1.10 had this characteristic and 213.198.77.122 had a TTL that was also a seven-bit value. There were changes in some of the routes over the period of 2013. For instance, the CAIDA traceroute to 194.71.107.251 on 22 April 2013 contained a 17-hop series. The next traceroute on 24 November 2013 contained an 18-hop series with the additional hop of 31.172.1.10 as shown in Figure 4.13. Another instance of varying results for a destination IP address was the traceroutes for 194.71.107.241. In this case, a 17-hop series was found dating back to 1 June 2013 while an 18-hop series was identified on 6th December 2013. We identified five of such occurrences where there was a transition from 17-hop series to the 18-hop series that were present in traceroutes obtained at a later date (i.e., we then observed an extra hop in deceptive portion of the traceroute).

A further inspection of historic CAIDA traceroutes to the 194.71.107.0/24 indicated that 195.69.147.245 was another major IP address involved in the deception. We counted 46 instances with a 17-hop series, where 195.69.147.245 replaced 213.198.77.122 as the sole hop before the long IP series of hops. Similarly, there were two more instances with 195.69.147.245 that had 18-hop series and they included 31.172.1.10 as the first hop in the series of hops. With the historic CAIDA DNS dataset, 195.69.147.245 was resolved to ams-ix.as39138.net. The discovery of a second major IP address involved in the deception process led us to continue analyzing the 194.71.107.0/24 subnet. We noticed a small number of IP addresses such as 184.105.213.101 and 194.146.118.105

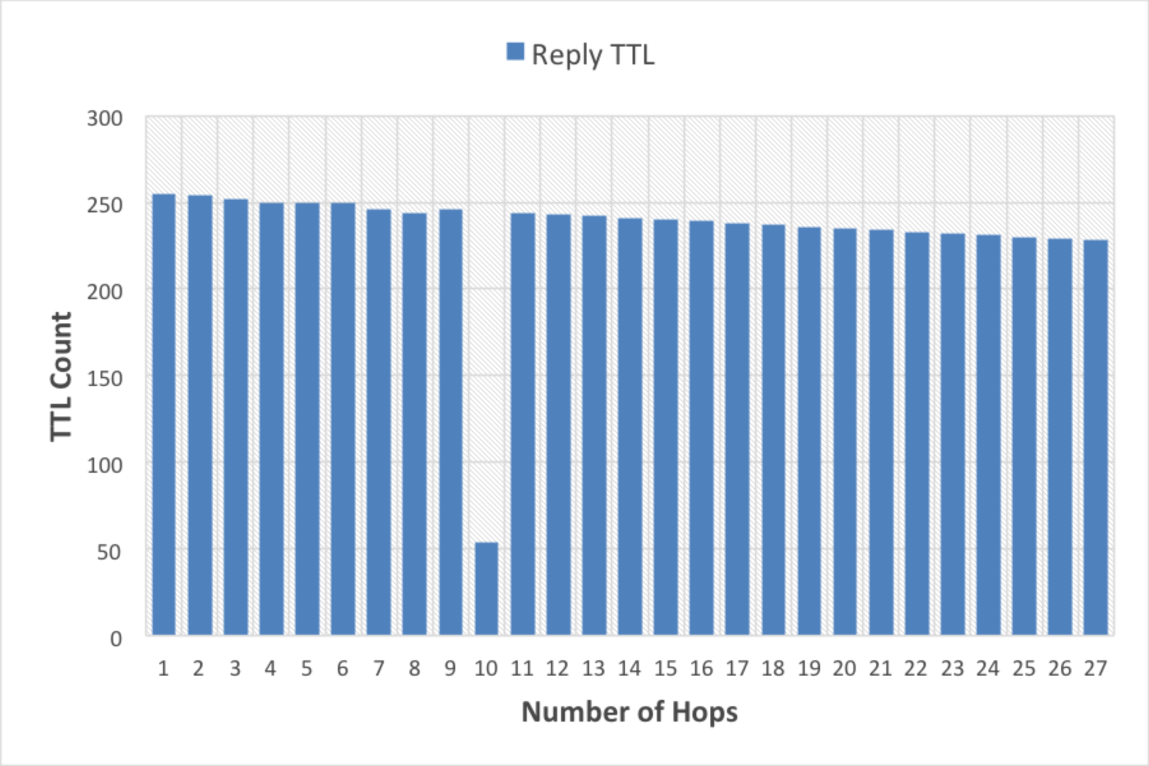


Figure 4.14: An instance of reply TTLs for traceroutes that end with 175.45.177.217.

that preceded the 17-hop series. In every traceroute that ended with the North Korean IP address 175.45.177.217, there would be a response probe that contained a reply TTL that consisted of two digits, as shown in Figure 4.14. The common 17-hop series meant that a fixed network topology was presented to incoming probes. It also revealed that probes from multiple vantage points collected the same network topology regardless of their actual geographical distance and that the route required a traversal through Europe and the North American continent.

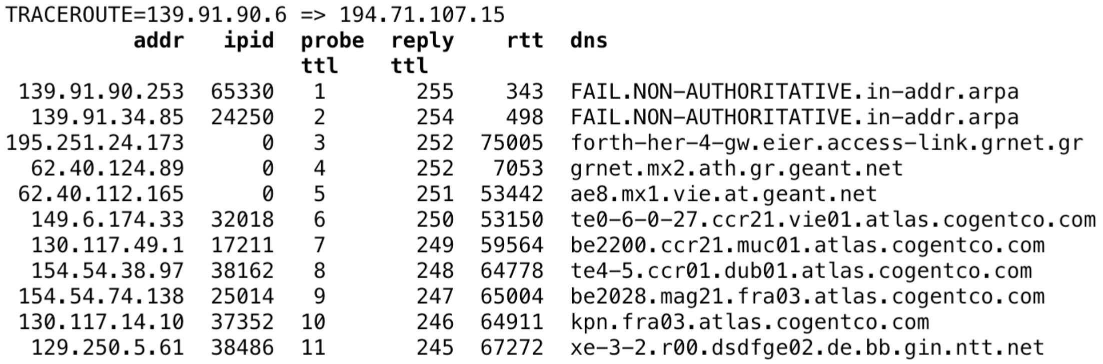
As a result of The Pirate Bay service being taken down [46], we were unable to probe the service directly for more details. Our next step was to ascertain whether the historic CAIDA traceroute to 175.45.177.217 had any similarities with the 194.71.107.0/24 traceroutes. 175.45.177.217 was the IP address in North Korea which was the last responding hop in traceroutes to The Pirate Bay prior to its demise. From Figure 4.15, the 175.45.177.217 did not contain any of the hops in the common 17-hop series that existed in traceroutes to IP addresses in the 194.71.107.0/24 subnet. Given that a direct traceroute showed a completely separate route, it would have made the inferred topology based on the 17-hop series highly suspicious even if we did not know about the actual deceit in the first place.

```
TRACEROUTE=192.231.228.5 => 175.45.177.217
          addr  ipid   probe  reply      rtt  dns
                       ttl    ttl
  192.231.228.1      0      1     255      145  FAIL.SERVER-FAILURE.in-addr.arpa
209.87.254.213  30442      2     254     1120  core-tunnel97.storm.ca
209.87.239.161  46194      3     253     1125  core-2-g0-3.storm.ca
 67.69.228.161      0      4     252     2005  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 64.230.164.17      0      5     251     1746  core2-ottawa23\x5fpos13-1-0.net.bell.ca
 64.230.99.250  50067      6     246    22813  tcore4-ottawa23\x5f0-4-2-0.net.bell.ca
 64.230.79.222  61476      7     245    20841  tcore3-montreal01\x5fpos0-14-0-0.net.bell.ca
 64.230.32.145      0      8     245    20707  bx4-montrealak\x5fxe2-0-0\x5f0.net.bell.ca
64.230.147.222  26885      9     247    29763  bx1-ashburn\x5fso-4-1-0.net.bell.ca
64.230.185.137      0     10     245    18792  bx2-ashburn\x5fxe2-1-0.net.bell.ca
    12.89.71.9      0     11     243    22968  FAIL.NON-AUTHORITATIVE.in-addr.arpa
12.122.113.102  63244     12     233    89983  cr81.mpsmn.ip.att.net
  12.122.18.21   6647     13     234    89070  cr1.cgcil.ip.att.net
  12.122.31.85  45576     14     235    86698  cr2.dvmco.ip.att.net
  12.122.30.25   9181     15     236    86264  cr1.slkut.ip.att.net
  12.122.30.30  55078     16     237    89879  cr2.la2ca.ip.att.net
 12.123.30.249  10353     17     238    89927  cr84.la2ca.ip.att.net
12.122.129.117      0     18     239    85895  gar1.lsrca.ip.att.net
   12.119.9.50      0     19     238   248977  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 219.158.39.38  14484     20     237   248796  FAIL.NON-AUTHORITATIVE.in-addr.arpa
209.87.254.213  22461     25     237   405764  core-tunnel97.storm.ca
```

Figure 4.15: Traceroute to 175.45.177.217 on 5 July 2013 (daily.l7.t2.c002608.20130705.yow-ca.warts.gz) in the 2013 Dataset.

4.3 Star Wars

As the Star Wars traceroute was no longer operational, we could only rely on historic traceroutes from the CAIDA IPv4 Routed /24 Topology Dataset. Based on the 2013 dataset, we observed the characteristics of traceroute probes bouncing between two routers with a VRF setup. There was a fixed series of 47 hops whose IP addresses fell within the same 206.214.251.0/24 subnet. There were three main patterns that stood out in this traceroute: a) incremental IP addresses, b) dual incremental IPID sequences, and c) duplicate TTL values. The IP addresses appeared in the traceroute responses in an ascending order beginning from 206.214.251.1. The IP address sequence also alternated between incrementing by a value of three and five for each hop. We considered two separate hop streams with alternating IP addresses and IPID flows. The first stream began at 206.214.251.1



```
TRACEROUTE=139.91.90.6 => 194.71.107.15
         addr   ipid  probe  reply    rtt  dns
                      ttl    ttl
  139.91.90.253  65330   1      255    343  FAIL.NON-AUTHORITATIVE.in-addr.arpa
   139.91.34.85  24250   2      254    498  FAIL.NON-AUTHORITATIVE.in-addr.arpa
 195.251.24.173      0   3      252  75005  forth-her-4-gw.eier.access-link.grnet.gr
   62.40.124.89      0   4      252   7053  grnet.mx2.ath.gr.geant.net
  62.40.112.165      0   5      251  53442  ae8.mx1.vie.at.geant.net
   149.6.174.33  32018   6      250  53150  te0-6-0-27.ccr21.vie01.atlas.cogentco.com
   130.117.49.1  17211   7      249  59564  be2200.ccr21.muc01.atlas.cogentco.com
   154.54.38.97  38162   8      248  64778  te4-5.ccr01.dub01.atlas.cogentco.com
  154.54.74.138  25014   9      247  65004  be2028.mag21.fra03.atlas.cogentco.com
  130.117.14.10  37352  10      246  64911  kpn.fra03.atlas.cogentco.com
   129.250.5.61  38486  11      245  67272  xe-3-2.r00.dsdfge02.de.bb.gin.ntt.net
```

Figure 4.16: Traceroute to The Pirate Bay (194.71.107.15) on 22 February 2014 (daily.l7.t3.c003012.20140222.her-gr.warts.gz) in the 2014 Dataset.

with subsequent hops increasing the last octet value by eight. Its IPID sequence started from 58775 and increased between one and five values for every subsequent hop. Similarly, the second stream began at 205.214.251.6 with the same increasing last octet by eight, and its IPID sequence started from 48277 onwards with an increase of between one and four for every next hop. As we could not probe the system for further information, we suggest that the randomness in the IPID advancement could be attributed to the routers replying to other external probes at the same time. These two streams corresponded with the published inner workings of the gag in the Star Wars author's blog [3]. The reply TTL values were equivalent throughout the two-stream chain as a result of the default route in his original design. Lastly, the RTT had a range between 194 ms and 207 ms and a standard deviation of 4 ms for the hops in the 206.214.251.0/24 subnet. The short RTT presented a reasonable picture of an internal network.

We observed the same alternating patterns in all four historic traceroutes from the 2014 dataset, as well as three from the ongoing 2015 datasets. Our original methodology did not search for any potential traceroutes of this alternating IPID nature in the historic CAIDA datasets. As such, the Star Wars traceroutes were completely ignored in our filtering of the datasets. The traceroutes in 2014 and 2015 datasets too had varying randomness in their IPID advancements. Figure 4.18 shows the change in IPID values for each of the two alternating hop streams we considered in one of the Star Wars traceroutes in the 2014 dataset. The chart indicated no discernible pattern other than that the IPIDs were increasing mono-

```
TRACEROUTE=196.200.131.131 => 216.81.59.173
          addr   ipid  probe  reply     rtt  dns
                         ttl    ttl
196.200.131.129  61842      1    255     282  gw.cnrst.ma
196.200.131.133  31796      2    255     252  gw.marwan.ma
196.200.163.253      0      3    253    1463  cnrst-mw.ll.marwan.ma
  130.117.3.122  16768      8    249   40434  te2-1.ccr01.mad03.atlas.cogentco.com
   77.67.72.125      0      9    247   40192  ae0-10.mad44.ip4.tinet.net
141.136.108.134      0     10    241  174077  xe-4-3-0.atl11.ip4.tinet.net
   77.67.69.158  58463     11    241  170581  epik-networks-gw.ip4.tinet.net
  206.214.251.1  58775     13    239  201604  episode.iv
  206.214.251.6  48277     14    239  195508  a.new.hope
  206.214.251.9  58777     15    239  202347  it.is.a.period.of.civil.war
 206.214.251.14  48279     16    239  194240  rebel.spaceships
 206.214.251.17  58778     17    239  203584  striking.from.a.hidden.base
 206.214.251.22  48280     18    239  196804  have.won.their.first.victory
 206.214.251.25  58779     19    239  203473  against.the.evil.galactic.empire
 206.214.251.30  48282     20    239  195800  during.the.battle
 206.214.251.33  58780     21    239  201981  rebel.spies.managed
 206.214.251.38  48283     22    239  195789  to.steal.secret.plans
 206.214.251.41  58782     23    239  204205  to.the.empires.ultimate.weapon
```

Figure 4.17: Partial Star Wars Traceroute (216.81.59.173) on 27th July 2013 (daily.l7.t2.c002647.20130727.cmn-ma.warts.gz) in the 2013 Dataset showing sequential and duplication patterns.

tonically. We did not find a constant difference in any of the Star Wars traceroutes in all datasets. The origin points in the 2014 dataset included China, Indonesia, and Spain. The use of multiple vantage points and their similar results concluded that there was no dynamic topology variance. Furthermore, as the subnet addresses belonged to Epik Networks [47], an ISP based in North America, there were no telltale signs of unfeasible routes from other continents like the Pirate Bay example in Section 4.2. Another noteworthy point was that the IP addresses used belonged to an unused, non-routed /24 network for which the gag creator had access to. The gag did not require IP address spoofing, unlike The Pirate Bay case. This access not only allowed Werber to generate reverse DNS names based on the Star Wars theme, but also to leverage upon their inaccessibility. Hence, pings to the IP addresses in the `206.214.251.0/24` subnet had no responses and the corresponding CAIDA traceroutes did not show a route through the `206.214.251.*` address used in the gag.

Interestingly, the last responding hop at the end of each historic Star Wars traceroute in all datasets had a distinct IP address. For example, one traceroute conducted on 25th July 2014 yielded a `*.comcast.net` address, and another traceroute on 19th January 2015 had a `*.corp.gq1.yahoo.com` address as seen in Figure 4.19. By performing a *whois* lookup

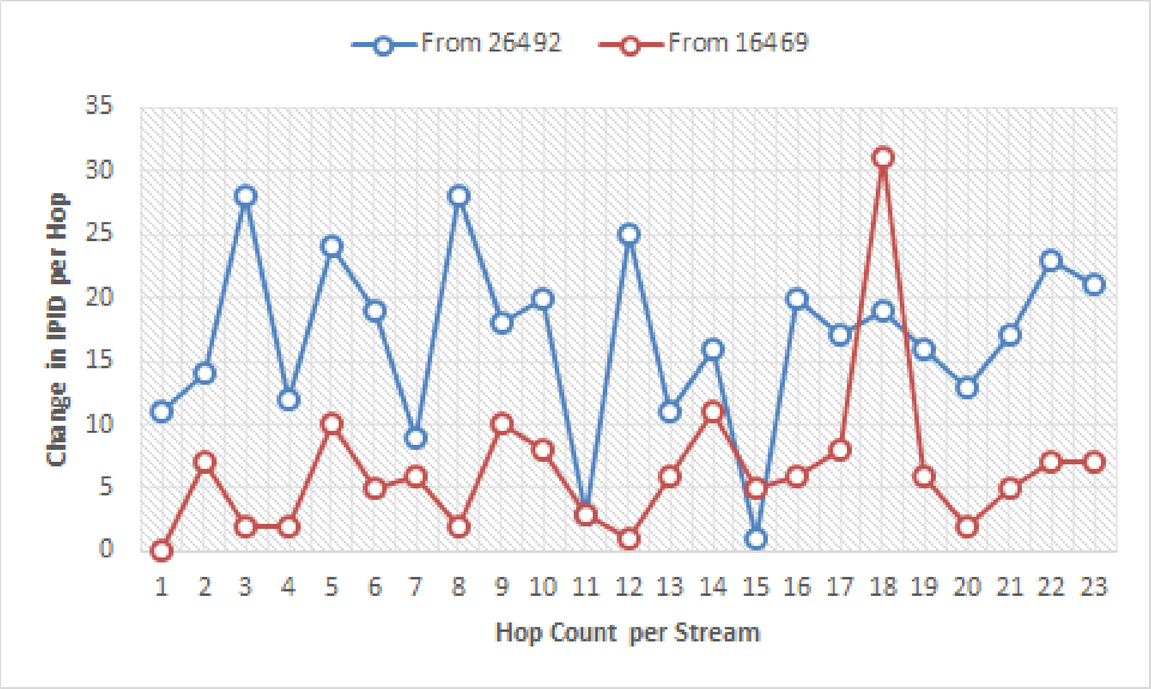


Figure 4.18: Change in IPID values for each hop stream in a Star Wars traceroute to 216.81.59.173 on 16th December 2014 (daily.l7.t3.c003700.20141216.she-cn.warts.gz) in the 2014 dataset.

on the last responding hop addresses of the historic Star Wars traceroutes in all datasets, we found that the IP addresses were unique and belonged to organizations in the United States, China, Macau, and Bosnia. Although the Star Wars opening crawl was the original intriguing feature of this case, we postulate that the peculiarity of the last hop might suggest a redirection to a random IP address, and it could be worth a future investigation.

4.4 Inside the Christmas Carol

The Christmas Carol traceroute was advertised as a holiday themed gag by MyssT [7] on Christmas Eve in 2014 some time after the initial release of the Star Wars themed traceroute gag. Therefore, we were not able to obtain traceroutes containing the spoofed topology in the 2013 and 2014 datasets. However, the gag is still operational and we are able to probe it directly. Unlike the Star Wars traceroute gag, there was no blog or article that described the Christmas Carol traceroute's inner workings. Instead, we discovered an Internet Relay Chat

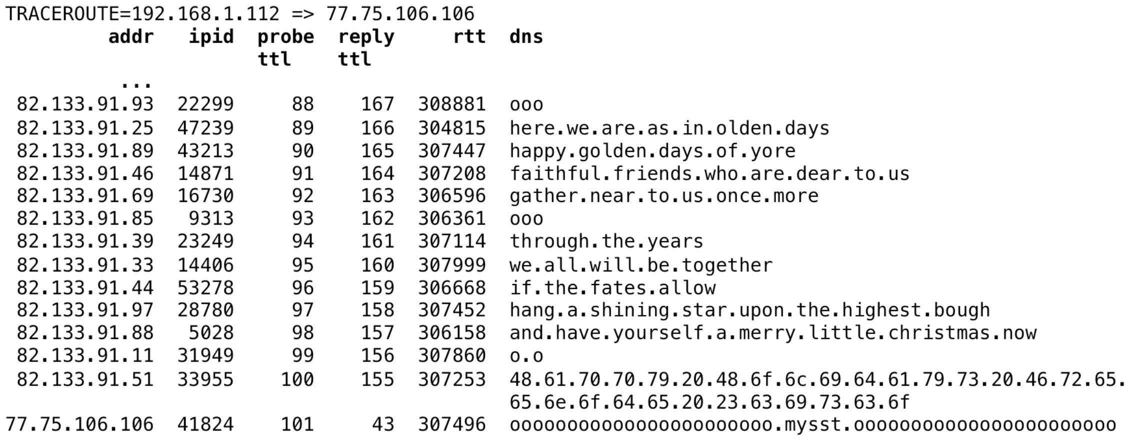
```
TRACEROUTE=192.172.226.247 => 216.81.59.173
           addr   ipid  probe  reply     rtt  dns
                        ttl    ttl
            ...
206.214.251.166  58881     53    243  113087  0-0
206.214.251.169  39749     54    243  112581  00
206.214.251.174  58883     55    243  113356  i
206.214.251.177  39750     56    243  107735  by.ryan.werber
206.214.251.182  58884     57    243  113151  blizzards.breed.ccie.creativity
206.214.251.185  39752     58    243  113492  please.try.again.tracerote.to.obiwan.scrye.net
 98.137.126.131     59     59    243  109487  hv02.cal.corp.gq1.yahoo.com


TRACEROUTE=192.172.226.247 => 98.137.126.131
           addr   ipid  probe  reply    rtt    dns
                        ttl    ttl
            ...
216.115.101.111      0      9    245  43564  ae-5.pat1.gqb.yahoo.com
    66.196.67.3      0     10    245  43589  ae-1.msr2.gq1.yahoo.com
   67.195.1.161  29343     11    244  43033  te-3-1.cer1.corp.gq1.yahoo.com
   98.136.200.6  20909     12    242  43948  re-0.cfw-a-gci.corp.gq1.yahoo.com
  98.136.200.67  40712     13    242  43577  vl-947.clr2-a-gci.corp.gq1.yahoo.com
 98.136.200.137     14     14    241  44892  po-263.bas1-1-gci.corp.gq1.yahoo.com
 98.137.126.131  12455     15     49  44296  hv02.cal.corp.gq1.yahoo.com
```

Figure 4.19: Comparison of last seven hops to 216.81.59.173 on 19th January 2015 (daily.l7.t1.c003767.20150119.san-us.warts.gz) in the 2015 dataset.

(IRC) archive which supposedly contains the creator's comments that the fake topology was implemented using *iptables* [48] and Perl extensions instead of the Star Wars traceroute's original VRF design [49]. As such, we would not expect to see alternating hop streams with duplicated TTL values.

Figure 4.20 shows the last 14 hops of the operational Christmas Carol traceroute on 11 June 2015. We performed a series of traceroutes based on options available in *Scamper* such as UDP, UDP-Paris, ICMP, ICMP-Paris, TCP, and TCP-ACK. We also extracted a historic CAIDA traceroute to `77.75.106.106` on 28 March 2015. As we were aware that there would be a chain of IP addresses within a certain subnet, we quickly located a series of continuous hops in the general `82.133.91.0/25` block, starting from `77.75.106.106` and ending with `82.133.91.51`. A *whois* lookup showed that the `82.133.91.0/24` block was owned by Tiscali UK, now known as TalkTalk, a telecommunications company [50]. There were 80 hops in the subnet and the RTTs were averaging around 164 ms with a standard deviation of 2 ms. The IPID sequences did not exhibit any obvious pattern like the alternating or consecutive values. Similarly, the TTLs were decrementing accordingly. The traceroutes ended with the original destination `77.75.106.106`. We observed that



```
TRACEROUTE=192.168.1.112 => 77.75.106.106
        addr   ipid  probe  reply     rtt  dns
                      ttl    ttl
         ...
  82.133.91.93  22299     88    167  308881  ooo
  82.133.91.25  47239     89    166  304815  here.we.are.as.in.olden.days
  82.133.91.89  43213     90    165  307447  happy.golden.days.of.yore
  82.133.91.46  14871     91    164  307208  faithful.friends.who.are.dear.to.us
  82.133.91.69  16730     92    163  306596  gather.near.to.us.once.more
  82.133.91.85   9313     93    162  306361  ooo
  82.133.91.39  23249     94    161  307114  through.the.years
  82.133.91.33  14406     95    160  307999  we.all.will.be.together
  82.133.91.44  53278     96    159  306668  if.the.fates.allow
  82.133.91.97  28780     97    158  307452  hang.a.shining.star.upon.the.highest.bough
  82.133.91.88   5028     98    157  306158  and.have.yourself.a.merry.little.christmas.now
  82.133.91.11  31949     99    156  307860  o.o
  82.133.91.51  33955    100    155  307253  48.61.70.70.79.20.48.6f.6c.69.64.61.79.73.20.46.72.65.
                                           65.6e.6f.64.65.20.23.63.69.73.63.6f
77.75.106.106  41824    101     43  307496  ooooooooooooooooooooooo.mysst.ooooooooooooooooooooooo
```

Figure 4.20: Partial traceroute to xmas.futile.net (77.75.106.106) on 11 June 2015.

while the chain containing the 82.133.91.0/25 block had three-digit reply TTL values, the preceding and final hops had two-digit TTLs.

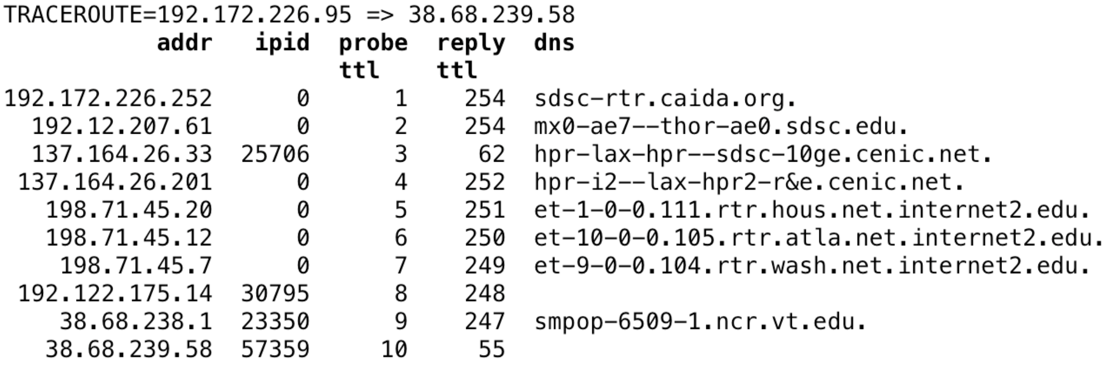
```
TRACEROUTE=192.168.1.112 => 77.75.106.106
         addr   ipid  probe  reply  rtt     aspath
                        ttl    ttl
          ...
  93.89.84.75  44972     20     44  306767  39326 [39326 9105 Not Found]
 82.133.91.37    212     21    234  327331  9105 [Same AS]
 82.133.91.18  57600     22    233  287368  9105 [Same AS]
          ...
 82.133.91.11  31949     99    156  307860  9105 [Same AS]
 82.133.91.51  33955    100    155  307253  9105 [9105 39326 Not Found]
77.75.106.106  41824    101     43  307496  39326
```

Figure 4.21: Partial traceroute to xmas.futile.net (77.75.106.106) with matching AS links on 11th June 2015.

We performed traceroutes from both the North and South American continents and noticed the same fixed route through the 82.133.91.0/25 subnet. This fixed route meant that we could not utilize the technique of finding routing circuits that was described in Section 3.1.2. Circuits in traceroutes suggest a routing inefficiency or deliberate manipulation. Compared with the Star Wars traceroute, multiple vantage points did not contribute much as every last hop's IP address was identical throughout. Our pings and traceroutes to the IP addresses in the subnet involved returned non-responses. A possible reason would be the creator's use of unused IP addresses as seen earlier in the Star Wars traceroute. Figure 4.21 shows the results of our AS matching. It found that based on the RIB records, there were no AS links between the hop before the subnet chain and the subnet chain itself, as well as the subnet chain and the last hop. The lack of direct AS links suggested that the route was not real.

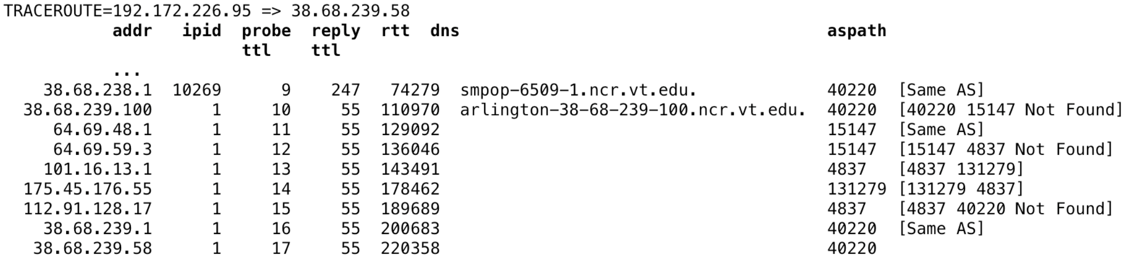
4.5 Unmasking the DeTracer

The DeTracer's goal was to present an arbitrary false network topology to an adversary trying to map the defender's network [33]. The DeTracer was operational for a period of time before its infrastructure was taken down. The DeTracer could respond to probes by presenting a static network topology or generating random IP addresses for each distinct traceroute. While the DeTracer was configured to present a static topology, we probed it with *Scamper* using ICMP, UDP, and TCP. Figure 4.22 shows the normal route generated when the deception feature was disabled.



```
TRACEROUTE=192.172.226.95 => 38.68.239.58
           addr   ipid  probe  reply  dns
                        ttl    ttl
192.172.226.252      0      1    254  sdsc-rtr.caida.org.
  192.12.207.61      0      2    254  mx0-ae7--thor-ae0.sdsc.edu.
  137.164.26.33  25706      3     62  hpr-lax-hpr--sdsc-10ge.cenic.net.
 137.164.26.201      0      4    252  hpr-i2--lax-hpr2-r&e.cenic.net.
   198.71.45.20      0      5    251  et-1-0-0.111.rtr.hous.net.internet2.edu.
   198.71.45.12      0      6    250  et-10-0-0.105.rtr.atla.net.internet2.edu.
    198.71.45.7      0      7    249  et-9-0-0.104.rtr.wash.net.internet2.edu.
 192.122.175.14  30795      8    248
    38.68.238.1  23350      9    247  smpop-6509-1.ncr.vt.edu.
   38.68.239.58  57359     10     55
```

Figure 4.22: Traceroute to 38.68.239.58 with no deception in place.



```
TRACEROUTE=192.172.226.95 => 38.68.239.58
          addr   ipid  probe  reply  rtt  dns                                   aspath
                       ttl    ttl
           ...
   38.68.238.1  10269      9    247   74279  smpop-6509-1.ncr.vt.edu.           40220  [Same AS]
 38.68.239.100      1     10     55  110970  arlington-38-68-239-100.ncr.vt.edu.  40220  [40220 15147 Not Found]
    64.69.48.1      1     11     55  129092                                     15147  [Same AS]
    64.69.59.3      1     12     55  136046                                     15147  [15147 4837 Not Found]
   101.16.13.1      1     13     55  143491                                     4837   [4837 131279]
 175.45.176.55      1     14     55  178462                                     131279 [131279 4837]
 112.91.128.17      1     15     55  189689                                     4837   [4837 40220 Not Found]
   38.68.239.1      1     16     55  200683                                     40220  [Same AS]
  38.68.239.58      1     17     55  220358                                     40220
```

Figure 4.23: Partial traceroute to 38.68.239.58 with fixed deceptive topology enabled.

Figure 4.23 is an example of a fixed topology deception. All our probes reported that an identical topology based on the source IP addresses of the return packets. The deception was observed to begin after 38.68.239.100, and contained obvious patterns such as identical IPID and TTL values. The deception hop chain had spoofed IP addresses belonging to various destinations. For instance, *whois* lookups showed that the probes traversed through 38.68.239.100 (Virginia Polytechnic Institute and State University), followed by 64.69.48.1 (U.S. Department of Homeland Security), 101.16.13.1 (China Unicom), 175.45.176.55 (Star Joint Venture Co.), 112.91.128.17 (China Unicom), and back to the Virginia Tech. Our AS path matcher determined that although there was a link between the China Unicom and North Korean IP addresses, there were no direct connections to the North American IP addresses before and after. This was in conjunction with the original deception topology as covered by West [33]. If the route contained additional hops whose IP addresses could bridge the broken AS links, a casual observer only looking at the AS links alone might have accepted the deceptive topology as the truth. Compared to the Star Wars traceroute in Section 4.3 and the Christmas Carol traceroute in Section 4.4, the use of spoofed IP addresses in the Pirate Bay traceroute was similar to the DeTracer's. It also allowed us to probe the the individual IP addresses and discover conflicting topologies. If the choice of IP addresses resulted in a looped cross-continent route even when multiple vantage points were used, it would then indicate a manipulated route.

Figure 4.24 demonstrates the outcome of the DeTracer configured to spoof random IP addresses in its replies to incoming probes. Each of our traceroutes resulted in a different set of IP addresses in its deception chain. We were able to recognize the duplicated IPID and TTL values. Tellingly, the IPID values in the deception chain for the both the static and dynamic topologies were identical and always a value of one. Another obvious pattern would be the RTTs advancing steadily from 149 ms to 588 ms in the deception chain, only to revert to a low RTT of 88 ms in the last hop. This was unlike the fixed topology configuration where the RTT did not change drastically in the last hop. As the IP addresses were based on a random selection, it was trivial to detect manipulation due to the geographical haphazardness of the route presented.

```
TRACEROUTE=192.168.1.1 => 38.68.239.50
           addr   ipid  reply  rtt      dns
                        ttl
            ...
   38.68.238.13  23256    243   89 ms   new-mag-smpop.ncr.vt.edu
190.118.141.170      1     51  142 ms
  81.13.114.194      1     51  161 ms   voynova-mn.rmt.ru
183.228.165.212      1     51  170 ms
109.176.228.102      1     51  201 ms
  144.90.243.85      1     51  231 ms
  79.193.75.102      1     51  310 ms   p4FC14B66.dip0.t-ipconnect.de
173.191.156.237      1     51  260 ms   h237.156.191.173.dynamic.ip.windstream.net
 172.185.254.65      1     51  266 ms   ACB9FE41.ipt.aol.com
 170.56.236.139      1     51  282 ms
  179.225.28.22      1     51  292 ms   179-225-28-22.user.vivozap.com.br
  88.205.63.147      1     51  312 ms
 198.106.37.149      1     51  341 ms
 222.213.19.123      1     51  340 ms
 108.216.203.35      1     51  366 ms   108-216-203-35.lightspeed.okcbok.sbcglobal.net
  76.139.180.95      1     51  403 ms   m2caba451bd33.chlm4.ma.comcast.net
  139.152.23.40      1     51  390 ms
 75.177.123.183      1     51  411 ms   cpe-075-177-123-183.triad.res.rr.com
105.108.105.176      1     51  426 ms
 105.235.40.137      1     51  436 ms   host-105-235-40-137.afnet.net
  50.19.239.117      1     51  449 ms   ec2-50-19-239-117.compute-1.amazonaws.com
   120.61.14.75      1     51  480 ms   triband-mum-120.61.14.75.mtnl.net.in
  40.123.45.220      1     51  483 ms
 187.73.254.154      1     51  499 ms   host-187-73-254-154.consoftmg.com.br
   74.6.213.179      1     51  513 ms   UNKNOWN-74-6-213-X.yahoo.com
  73.212.77.109      1     51  525 ms
101.110.137.166      1     51  552 ms   kyoei-corporation.co.jp
    1.229.21.27      1     51  560 ms
   75.104.25.12      1     51  588 ms   75-104-25-12.cust.wildblue.net
   38.68.239.50      1     51   88 ms
```

Figure 4.24: Partial traceroute to 38.68.239.50 with random topology enabled.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5: Conclusion

The case studies presented us with examples of traceroute manipulation of varying strengths and weaknesses in their implementation. Those weaknesses included inconsistencies stemming from the incompleteness of the desired deception strategy as well as the inability to adapt to probes originating from different vantage locations. The strengths highlighted the means whereby topology deception could be more effective and hinder our detection progress.

The Pirate Bay example saw an adversary attempting to portray a fake topology and adapting to flaws in the perceived route by correcting the impossible AS links in the last two hops with a Chinese IP address in place of the original infeasible Cambodian IP address. It did not display any obvious IPID and TTL patterns which would have allowed us to automatically discover it in the IPv4 Routed /24 Topology Dataset using our aforementioned methodology. Fortunately, a direct traceroute to the last responsive hop enabled us to detect an anomaly in the presented topology. The anomaly was that the presented topology had a North Korean host in its last responding hop, while the other CAIDA traceroutes to the said North Korean host yielded a totally different topology as the IP addresses were different. Likewise, the collection of historic traceroutes originating from multiple geographically distributed locations aided our analysis by showing that the routing was always the same regardless of the vantage point. For instance, a prober based in China would observe a traceroute path with a loop through Europe and North America, before returning to China—a path that is unlikely in normal routing circumstances. We might use this technique to identify such anomalous traceroutes automatically in future work.

The Star Wars gag using the Cisco VRF setup was presented as IPIDs drawn alternatively from two monotonic counters, as an artifact of using two physical routers to implement the deception. Our base filters do not account for this pattern, and it would be useful for a future filter to do so. While the Star Wars implementation utilized two hardware routers, an ideal filter should be able to detect any number of devices/hops with a common IPID counter without any limitation on the number of counters. The duplicate TTL values, the long chain of hops in a common subnet, and non-responsive IP addresses used in the reply hops are potential oddities that raise our interest in inspecting the inferred topology further.

Building on the Star Wars gag's method of utilizing pre-owned and unused IP addresses, the Christmas Carol gag did not present obvious signs of manipulation such as increasing IPIDs and duplicate TTLs. However, it retained the inconsistencies of non-responsive IP addresses in its common subnet chain and broken AS links before and after that chain of hops. The non-responsive IP addresses did not respond to ping requests and if there were any CAIDA traceroutes to those addresses, it did not show a route through the `206.214.251.*` address used in the gag. The broken AS links indicate that the relationship between the affected AS numbers did not exist in the RIB records we obtained from the Route Views Archive Project. Compared to the Pirate Bay case, the fact that the deceptive hop chain occurred solely within the United Kingdom prevented us from leveraging our worldwide distributed probers and discovering if there were inefficient cross-continental links. If the machine generating the deceptive hop chain was in the same country as the deception itself, it would mean that probers based in a foreign country would not have been able to identify additional continents for the inefficiency check. The gag's ability to respond equally to various probe types with the desired topology prevented us from uncovering any discrepancies resulting from an incomplete deception implementation. An incomplete deception implementation might present a fake topology for a specific traceroute probe type such as UDP and allow the system default implementation for a regular ICMP traceroute.

The case of the DeTracer identified flaws in implementation of an experimental topology deception system such as the duplicate TTLs, identical IPIDs for all traceroutes, and contrasting differences in the RTT for the last two hops. While the random IP addresses allowed for deception detection due to the possibility of broken AS links in-between hops, the experiment showcased the potential of constructing a realistic fake topology similar to the Pirate Bay's when probed from a location in Europe. It also showed that adversaries could craft a route that eliminated inconsistencies in IPIDs, TTLs, AS links, and RTT values. If the fake topology was deployed as part of an internal network, it would be harder to unravel deception involved as we would be unable to make use of our multiple vantage points.

Table 5.1 shows the summary of inconsistencies we exploited to raise the possibility of

Case Study	Duplicate TTL	Consecutive IPID	Broken AS Link	RTT Inconsistencies	Non-responsive Hops	Geographical Inconsistencies
Pirate Bay	No	No	No	Yes	No	Yes
Star Wars	Yes	Yes	Yes	No	Yes	No
Christmas Carol	No	No	Yes	No	Yes	No
DeTracer	Yes	No	Yes	Yes	No	Yes

Table 5.1: Summary of inconsistencies found in case studies.

deception in the case studies.

In our filtered IPv4 Routed /24 Topology Dataset, we had results suggesting anomalies based on our IPID and TTL filters. There were large numbers of traceroutes with duplicate TTL values as well as long consecutive IPID chains that we termed Akamai, Korean Telecom and China Telecom patterns. We contacted Akamai Technologies but the enquiry was inconclusive with regards to the consecutive IPID chains present in numerous varying traceroutes. There is a possibility that the DNS PTRs were out-of-date and not reflective of actual Akamai equipment and ownership. Another possibility was the spoofing of IP addresses by a misconfigured machine which used to own the IP address that was resolved to an Akamai hostname. If we were to compare the filtered traceroutes in the 2013 dataset with their counterparts in the 2014 dataset, the consecutive IPID chains of 2013 would morph into monotonically increasing IPID chains of varying step increments instead, as evidenced in the 2014 Korean Telecom pattern, or return to seemingly random IPID values in the newer China Telecom pattern. The earlier case studies showed that not all detection strategies were successful and more work would be required to programmatically narrow down the list of potential traceroutes with deceptive inferred topologies.

5.1 Future Work

The first known limitation in the original filter set is that only detecting instances of strictly continuous consecutive IPID chains is too restrictive. This limitation is evident in the case studies, such as the Star Wars example, and the matching traceroutes from 2013 in the 2014 dataset. The identification of IPID chains can be improved by catering for stepped increments, alternating hop streams, non-responsive hops, and duplicate IPIDs. Constant IPID chains in a traceroute might indicate the repeated use of a single machine in generating replies. As the use of constant IPID chains had been found in the DeTracer output, it could be the case where other deception systems might have this inconsistency. This may work in conjunction with a filter that detects gaps in the IPID values from one hop to the next by calculating the rate of change of the IPIDs. An added flexibility to specify a maximum gap limit will allow us to adapt to long IPID chains with varying gaps of different sizes. Finding monotonically increasing IPID sequences, and identifying multiple such streams, can give us an opportunity to automatically discover manipulated traceroutes like the Star Wars gag. In addition, a duplicate IPID count and its associated IPID value will enable us to detect traceroutes like the DeTracer's. By having more metrics to consider, we may then apply a weighted approach to rank and order potential suspicious traceroutes for further analysis. The duplicate TTL metric could also be part of this group. A higher weight could be associated with duplicate TTLs in a traceroute with interesting IPID chains. This would allow for another round of validation for traceroutes with duplicate TTLs similar to the DeTracer and Star Wars case studies. In addition to the duplicate TTLs, future work may include detection of non-monotonic reply TTLs and TTLs which merely follow a sequence but are not consecutive. The identification of these metrics would aid in customizing subsequent rounds of filtering for further analysis.

As discussed in Section 4.1.2, there were significant numbers of traceroutes containing continuous consecutive IPID chains of six hops and below. The filtering process could be extended to include these smaller chains together with the more comprehensive IPID filters. Furthermore, the matching of filtered traceroutes found in the 2013 dataset could be applied to both the 2014 and 2015 datasets for a three-year continuity check.

An area which may be improved is geolocation of the individual IP addresses for each hop of the traceroutes. For instance, a lookup table containing approximate RTTs between source and destination countries involved will aid in determining whether a hop's RTT is realistic or feasible. Suppose a traceroute hop has an IP address residing in the same country as the prober and its RTT is 100 ms, and there is another hop in the same traceroute containing an IP address belonging to a country across the globe with a similar 100 ms RTT, suspicions will be raised as the reported location is inconsistent with the recorded timing. The RTTs similarities would mean that the data traveled faster than light to achieve the lower-than-expected latency. This is highly infeasible to go faster than the speed of light. The Pirate Bay's choice of having IP addresses in two continents (i.e., North America and Asia) exposes a susceptibility where a lookup table may identify a continental hop cycle if the prober is based in Asia, especially if the prober and the destination are in the same country.

Filtering the IPv4 Routed /24 Topology Dataset is a time-intensive process due to the sheer number of traceroutes collected every cycle. For example, if it took approximately 25 seconds to parse and filter a *warts* file with 792 traceroutes and there were 5,313,384,228 traceroutes in the 2013 dataset, the operation would take 5.3 years to complete. In response to the duration required, we ran the file processing operations in parallel under a simple script. While we used *Python* to process the compressed *warts* files, a non-interpreted language like C could shorten the processing duration. A distributed compute cluster such as an Apache Hadoop-based system would also provide scalability and reduced processing times [51]. An alternative approach could be to identify the IP addresses hosting web services and collect historic and real-time traceroutes instead. The focus is to prioritize IP addresses with known web services in the hope that a deception similar to The Pirate Bay case may be uncovered earlier. This approach takes advantage of the ability to time the TCP connections and compare the reported traceroute RTT and the resultant direct TCP RTT. It will also provide tracking of changes to a website's topology by following the IP addresses from resolved DNS names. Subsequently, the IPID and duplicate TTL pattern tests may be applied to narrow down the number of potential cases containing inconsistencies for further analysis.

The matching process of identifying traceroutes in the 2013 dataset against their equivalents in the 2014 dataset involved using the destination IP addresses as the key criteria. As a result, we also obtained traceroutes with source IP addresses different from the original traceroute in the 2013 dataset. A different approach could be to use the same vantage point and obtain traceroutes to IP addresses in the same /24 subnet, as the desired destination IP address. This would enable discovery of patterns within the same /24 subnet as evidenced in The Pirate Bay case.

The Pirate Bay and DeTracer case studies were able to generate the desired deceptive topologies based on spoofing IP addresses on fabricated hops. This implies a dependency on the hosting network to allow source IP address spoofing. Networks banning such behavior will be only be able to generate topologies using IP addresses they own. This limits traceroute deception to internal network addresses. If there was an efficient method to determine whether a particular network allowed source IP spoofing, we could then flag the network as a plausible target with an inherent ability to portray fake topologies with non-owned addresses. Organizations with large networks such as ISPs may contain transit points due to peering agreements. These agreements may complicate matters by allowing IP addresses outside of an ISP's original network.

In addition to reply and probe TTLs, the quoted TTLs may provide clues as to whether hops in a traceroute may be hidden when there are hops with quoted TTL values of zero. The zero-TTL forwarding anomaly explained by Augustin *et al.* highlighted the possibility of duplicate IP addresses with consecutive IPID values as a result of a router misconfiguration. It may improve on our understanding during the traceroute analysis.

Lastly, the steady growth of Internet Protocol version 6 (IPv6) addresses, given the saturation of the IPv4 space, will eventually require extra effort to analyze these IPv6 traceroutes [52]. The similarity between the two IP versions is the common underlying technique of utilizing the ICMP Time Exceeded and Destination Unreachable messages for traceroutes to work. The major difference is the issue of the searchable address space. The IPv4 address space is 2^{32} while the IPv6 address space is 2^{128}. It will be more difficult and time consuming to filter through the entire IPv6 dataset. Hence, we will need a prioritization list and faster filter implementations for such future work.

APPENDIX:

A.1 Raw Summary of the 2013 Filtered Dataset

hops	lconsec	ldconsec	adconsec	aconsec	httl
1	64508982	38896366	41914644	68994014	47666721
2	4423018	301696	613279	4441020	13733703
3	759464	123055	130679	764845	4895758
4	73711	10244	11130	75132	2236425
5	44026	3911	4007	44791	1409844
6	26834	1057	1091	27386	1464274
7	30601	131	140	33096	792606
8	18152	3	4	18760	502030
9	38200	2	2	38565	267701
10	16286	1	3	16848	109876
11	14480	1	2	14755	45949
12	10838	0	0	11022	22807
13	6274	0	0	6396	8989
14	5699	0	0	5743	6713
15	10320	0	0	10422	11530
16	31413	0	0	31690	30818
17	5604	0	0	5665	8764
18	609	0	0	620	1723
19	223	0	0	229	629
20	1613	0	0	1619	1778
21	3	0	0	6	108
22	4	0	0	4	59
23	2	0	0	6	54
24	3	0	0	6	43
25	2	0	0	5	28
26	2	0	0	1	8
27	2	0	0	5	11
28	2	0	0	4	19
29	3	0	0	5	14
30	2	0	0	6	11
31	5	0	0	8	23
32	1	0	0	4	10
33	3	0	0	4	19
34	1	0	0	1	15
35	3	0	0	4	19
36	3	0	0	4	5
37	1	0	0	1	4
38	2	0	0	2	5
39	1	0	0	1	3
40	1	0	0	0	2
41	0	0	0	1	1
42	0	0	0	0	5
43	1	0	0	1	0
44	0	0	0	0	1
45	0	0	0	0	3
47	1	0	0	1	1
49	1	0	0	1	3
51	0	0	0	0	2
52	1	0	0	1	1
53	0	0	0	0	2

54	1	0	0	1	1
55	1	0	0	1	1
56	1	0	0	2	4
57	1	0	0	1	1
58	0	0	0	0	2
59	1	0	0	2	2
60	0	0	0	1	0
61	1	0	0	1	1
62	0	0	0	0	1
65	2	0	0	2	3
68	1	0	0	1	2
70	1	0	0	1	1
72	0	0	0	0	2
73	1	0	0	1	1
75	1	0	0	1	1
77	0	0	0	0	1
78	1	0	0	1	0
79	0	0	0	0	1
82	2	0	0	2	2
84	0	0	0	0	1
85	0	0	0	0	1
86	0	0	0	0	2
87	0	0	0	0	1
88	0	0	0	0	2
89	0	0	0	1	0
90	0	0	0	0	3
91	0	0	0	0	1
93	0	0	0	0	1
94	0	0	0	0	3
97	0	0	0	0	1
98	0	0	0	0	1
101	0	0	0	0	1
105	0	0	0	0	1
106	0	0	0	0	1
111	1	0	0	1	2
115	0	0	0	0	1
118	1	0	0	1	0
119	0	0	0	1	1
120	0	0	0	0	1
121	0	0	0	0	1
128	0	0	0	0	1
131	1	0	0	0	2
133	0	0	0	0	1
135	0	0	0	0	1
136	0	0	0	0	1
137	1	0	0	1	2
138	0	0	0	0	1
139	0	0	0	0	2
140	0	0	0	0	1
142	0	0	0	0	2
143	0	0	0	0	1
145	1	0	0	1	1
146	0	0	0	0	1
149	0	0	0	0	1
159	0	0	0	0	1
160	0	0	0	0	1
178	0	0	0	1	1
180	0	0	0	0	1
187	0	0	0	0	1
192	0	0	0	0	1
193	0	0	0	0	1

196	0	0	0	0	2
198	0	0	0	0	1
210	0	0	0	0	1
213	0	0	0	0	1
233	0	0	0	1	0
240	0	0	0	0	1
243	0	0	0	0	1
252	0	0	0	0	1
254	0	0	0	0	1
291	0	0	0	0	1
300	0	0	0	0	1
301	0	0	0	1	0
311	0	0	0	0	1
328	0	0	0	0	1
369	0	0	0	0	1
415	1	0	0	1	0
449	0	0	0	0	1
459	1	0	0	0	0
466	0	0	0	0	1
565	0	0	0	0	1
699	1	0	0	1	0
1155	0	0	0	0	1
2363	0	0	0	1	0
4067	0	0	0	0	1

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] ATIS Committee PRQC. (2015). ATIS Telecom Glossary. [Online]. Available: http://www.atis.org/glossary/definition.aspx?id=3516

[2] L. Spitzner, *Honeypots: Tracking Hackers*. Boston, MA: Addison-Wesley, 2002.

[3] R. Werber. (2013, Feb). Star Wars Traceroute. [Online]. Available: http://beaglenetworks.net/post/42707829171/star-wars-traceroute

[4] (2015). Virtual routing and forwarding. [Online]. Available: http://www.cisco.com/c/en/us/td/docs/net_mgmt/active_network_abstraction/3-7/reference/guide/ANARefGuide37/vrf.html

[5] T. Liston. (2015). Tom Liston talks about LaBrea. [Online]. Available: http://labrea.sourceforge.net/Intro-History.html

[6] L. Alt, R. Beverly, and A. Dainotti, "Uncovering network tarpits with degreaser," in *Proc. 30th Annu. Comput. Security Appl. Conf.*, Dec. 2014. [Online]. Available: http://www.rbeverly.net/research/papers/degreaser-acsac14.html

[7] RKBicknell. (2015). HAPPY HOLIDAYS! Have a traceroute! :D. [Online]. Available: http://www.reddit.com/r/networking/comments/2qb86s/

[8] I. Paul. (2013, Mar.). The Pirate Bay admits to North Korean hosting hoax. [Online]. Available: http://www.pcworld.com/article/2030073/the-pirate-bay-admits-to-north-korean-hosting-hoax.html

[9] Will's Blog. (2013, Mar.). The Pirate Bay – North Korean hosting? No, it's fake. [Online]. Available: https://rdns.im/the-pirate-bay-north-korean-hosting-no-its-fake

[10] Manpages. (2015). Manpage for traceroute. [Online]. Available: http://man.cx/?page=traceroute(8)

[11] Microsoft. (2015). How to use the Tracert command-line utility to troubleshoot TCP/IP problems in Windows. [Online]. Available: https://support.microsoft.com/en-us/kb/162326

[12] J. Postel. (1981, Sept.). Internet Control Message Protocol. [Online]. Available: https://tools.ietf.org/html/rfc792

[13] B. Augustin, X. Cuvellier, B. Orgogozo, T. F. F. Viger, C. M. M. Latapy, and R. Teixeira, "Avoiding traceroute anomalies with Paris traceroute," in *Proceedings of the 6th ACM Conference on Internet measurement*, 2006.

[14] J. Quittek, T. Zseby, B. Claise, and S. Zander. (2004). Requirements for IP Flow Information Export (IPFIX). [Online]. Available: https://tools.ietf.org/html/rfc3917

[15] J. Hawkinson and T. Bates, “Guidelines for creation, selection, and tegistration of an autonomous system (AS),” RFC 1930 (Best Current Practice), Internet Engineering Task Force, Mar. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1930.txt

[16] Y. Rekhter, T. Li, and S. Hares, “Border Gateway Protocol 4,” RFC 4271, Internet Engineering Task Force, Jan. 2006. [Online]. Available: https://tools.ietf.org/html/rfc4271

[17] CAIDA. (2015). About CAIDA. [Online]. Available: http://www.caida.org/home/about/

[18] CAIDA. (2015). Archipelago (Ark) Measurement Infrastructure. [Online]. Available: http://www.caida.org/projects/ark/

[19] CAIDA. (2015). The IPv4 Routed /24 Topology Dataset. [Online]. Available: http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml

[20] CAIDA. (2015). Scamper. [Online]. Available: https://www.caida.org/tools/measurement/scamper/

[21] M. Luckie, “Scamper: A scalable and extensible packet prober for active measurement of the internet,” in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*. New York, NY: ACM, 2010, pp. 239–245. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879171

[22] CAIDA. (2015). IPv4 Routed /24 DNS Names Dataset. [Online]. Available: http://www.caida.org/data/active/ipv4_dnsnames_dataset.xml

[23] D. Meyer. (2015). University of Oregon Route Views Archive Project. [Online]. Available: http://archive.routeviews.org

[24] G. Y. Cai, “Ip infrastructure geolocation,” master’s thesis, Naval Postgraduate School, Mar. 2015. [Online]. Available: http://hdl.handle.net/10945/45165

[25] L. Daigle, “WHOIS protocol specification,” RFC 3912, The Internet Society, Sept. 2004. [Online]. Available: https://tools.ietf.org/html/rfc3912

[26] R. Kissel. (2003, June). Glossary of key information security terms. [Online]. Available: https://tools.ietf.org/html/rfc792

[27] Y. JJ, “Defensive computer-security deception operations: processes, principles and techniques,” Ph.D. dissertation, N. Carolina State Univ., Raleigh, NC., 2006.

[28] M. H. Almeshekah and E. H. Spafford, “Planning and integrating deception into computer security defenses,” in *Proceedings of the 2014 Workshop on New Security Paradigms Workshop*. New York, NY: ACM, 2014, pp. 127–138. [Online]. Available: http://doi.acm.org/10.1145/2683467.2683482

[29] D. E. Denning, “Framework and principles for active cyber defense,” *Comput. Secur.*, vol. 40, pp. 108–113, Feb. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2013.11.004

[30] K. E. Heckman, M. J. Walsh, F. J. Stech, T. A. O’Boyle, S. R. Dicato, and A. F. Herber, “Active cyber defense with denial and deception: A cyber-wargame experiment,” *Comput. Secur.*, vol. 37, pp. 72–77, Sept. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2013.03.015

[31] E. Biersack, C. Callegari, and M. Matijasevic, *Data traffic monitoring and analysis: From measurement, classification, and anomaly detection to quality of experience*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013. [Online]. Available: https://books.google.com/books?id=fCS6BQAAQBAJ

[32] S. T. Trassare, “A technique for presenting a deceptive dynamic network topology,” master’s thesis, Naval Postgraduate School, Mar. 2013. [Online]. Available: http://hdl.handle.net/10945/32911

[33] A. West, “Toward a robust method of presenting a rich, interconnected deceptive network topology,” master’s thesis, Naval Postgraduate School, Mar. 2015. [Online]. Available: http://hdl.handle.net/10945/45271

[34] S. Trassare, R. Beverly, and D. Alderson, “A technique for network topology deception,” in *Proceedings of the Military Communications Conference (MILCOM)*, Nov. 2013. [Online]. Available: http://hdl.handle.net/10945/36486

[35] J. Touch, “Updated Specification of the IPv4 ID Field,” RFC 6864, Internet Engineering Task Force, Feb. 2013. [Online]. Available: https://tools.ietf.org/html/rfc6864

[36] S. M. Bellovin, “A technique for counting natted hosts,” in *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*. New York, NY: ACM, 2002, pp. 267–272. [Online]. Available: http://doi.acm.org/10.1145/637201.637243

[37] S. Cheshire. (1996, May). It’s the latency, stupid. [Online]. Available: https://rescomp.stanford.edu/~cheshire/rants/Latency.html

[38] A. Rosen. (2014, Dec.). It's alarmingly easy to take North Korea's internet offline. [Online]. Available: http://www.businessinsider.com/easy-to-take-north-koreas-internet-offline-2014-12

[39] CAIDA. (2015). IPv4 Routed /24 AS Links Dataset. [Online]. Available: http://www.caida.org/data/active/ipv4_routed_topology_aslinks_dataset.xml

[40] R. NCC. (2015). libBGPdump. [Online]. Available: https://bitbucket.org/ripencc/bgpdump/wiki/Home

[41] Ernesto. (2013, Mar.). The Pirate Bay "Moves" To North Korea. [Online]. Available: https://torrentfreak.com/the-pirate-bay-moves-to-north-korea-gets-virtual-asylum-130304/

[42] (2015). After being cut from Norway, The Pirate Bay returns from North Korea. [Online]. Available: https://www.reddit.com/r/technology/comments/19nb00/after_being_cut_from_norway_the_pirate_bay/

[43] W. Blog. (2013, Mar.). The Pirate Bay – North Korean hosting? No, it's fake. (P2). [Online]. Available: https://rdns.im/the-pirate-bay-north-korean-hosting-no-its-fake-p2

[44] E. Rosen, A. Viswanathan, and R. Callon. (2001, Jan.). Multiprotocol Label Switching Architecture. [Online]. Available: https://tools.ietf.org/html/rfc3031

[45] D. B., L. M., M. P., and P. J., "Revealing MPLS tunnels obscured from traceroute," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 42, no. 2, pp. 87–93, Apr. 2012.

[46] K. Zetter. (2014, Dec.). Pirate Bay has been raided and taken down: Here's what we know. [Online]. Available: http://www.wired.com/2014/12/pirate-bay-raided-taken-down/

[47] E. Networks. (2015). About Our Company. [Online]. Available: http://www.epiknetworks.com/about.php

[48] H. Eychenne. (2015). iptables(8) - Linux man page. [Online]. Available: http://linux.die.net/man/8/iptables

[49] (2015). #cisco IRC Archive 2014-12-24. [Online]. Available: http://www.corecompute.com/cisco/cisco_20141224.html

[50] A. Laughlin. (2010, Jan.). Tiscali TV rebrands as TalkTalk. [Online]. Available: http://www.digitalspy.co.uk/tech/news/a194586/tiscali-tv-rebrands-as-talktalk.html

[51] hadoop. (2015). Welcome to Apache$^{\text{TM}}$ Hadoop®! [Online]. Available: http://hadoop.apache.org/

[52] L. Colitti, S. H. Gunderson, E. Kline, and T. Refice, "Evaluating IPv6 adoption in the Internet," in *PAM 2010*, 2010. [Online]. Available: http://www.pam2010.ethz.ch/papers/full-length/15.pdf

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California